

# A Practical Approach to Lossy Joint Source-Channel Coding

Maria Fresia<sup>†</sup> and Giuseppe Caire<sup>★</sup>

February 1, 2008

<sup>†</sup> Maria Fresia, Communications and Remote Sensing Lab. Université catholique de Louvain, Belgium (E-mail: [fresia@tele.ucl.ac.be](mailto:fresia@tele.ucl.ac.be))

<sup>★</sup> Giuseppe Caire, Ming Hsieh Dept. of Electrical Engineering, University of Southern California, Los Angeles, CA, USA (E-mail: [caire@usc.edu](mailto:caire@usc.edu)).

## Abstract

This work is devoted to *practical* joint source channel coding. Although the proposed approach has more general scope, for the sake of clarity we focus on a specific application example, namely, the transmission of digital images over noisy binary-input output-symmetric channels. The basic building blocks of most state-of the art source coders are: 1) a linear transformation; 2) scalar quantization of the transform coefficients; 3) probability modeling of the sequence of quantization indices; 4) an entropy coding stage. We identify the weakness of the conventional separated source-channel coding approach in the catastrophic behavior of the entropy coding stage. Hence, we replace this stage with *linear* coding, that maps directly the sequence of redundant quantizer output symbols into a channel code-word. We show that this approach does not entail any loss of optimality in the asymptotic regime of large block length. However, in the practical regime of finite block length and low decoding complexity our approach yields very significant improvements. Furthermore, our scheme allows to retain the transform, quantization and probability modeling of current state-of the art source coders, that are carefully matched to the features of specific classes of sources. In our working example, we make use of “bit-planes” and “contexts” model defined by the JPEG2000 standard and we re-interpret the underlying probability model as a sequence of conditionally Markov sources. The Markov structure allows to derive a simple successive coding and decoding scheme, where the latter is based on iterative Belief Propagation. We provide a construction example of the proposed scheme based on punctured Turbo Codes and we demonstrate the gain over a conventional separated scheme by running extensive numerical experiments on test images.

# 1 Introduction

Shannon’s source-channel *separation principle* [1] states that, in the limit of large block-length and for a large class of communication setups, the optimal performance can be approached by independently designing the source coding and the channel coding schemes. Driven by the separation principle, modern communication systems have been developed according to a rather rigid layered architecture [2]: the source coding functions are essentially relegated to the application layer, at the top of the protocol stack, while the channel coding functions are located at the link and physical layers, at the bottom of the protocol stack. While a separated (layered) architecture has the unquestioned advantage of modular system design, allowing the convergence of a great variety of services on a common data network infrastructure<sup>1</sup>, there are cases where a Joint Source-Channel Coding (JSCC) approach is called for. On one hand, there exist several relevant multi-terminal settings where the separated approach is known to be suboptimal [1]. On the other hand, even in standard point to point channels where Separated Source-Channel Coding (SSCC) is asymptotically optimal, the use of independently designed source and channel codes may result in poor performance in the practical non-asymptotic regime of finite block length and low complexity encoding/decoding. This paper is concerned with this second case.

As a typical example, consider the transmission of digital images on a wireless channel.<sup>2</sup> Present systems treat the compressed image as a data packet that must be delivered error-free to the destination, despite the fact that, differently from other kind of data, an image may be represented at the destination within some tolerable distortion. In a system based on separation, the distortion is introduced uniquely by source coding and the underlying link and physical layers struggle to deliver the source-encoded bits error-free, by using a combination of channel coding and retransmissions. Another typical example is represented by terrestrial or satellite Digital Video Broadcasting [3, 4]. Here retransmissions cannot be used. Therefore, the channel coding scheme is designed to achieve a very strict BER target of  $10^{-10}$  or below. In both these examples, a very demanding performance requirement is imposed on the physical layer. This is due to the fact that the source coding scheme, that was designed assuming an error-free channel, exhibits a *catastrophic* behavior with respect to channel errors: even a small fraction of bits

---

<sup>1</sup>The modern evolution of Internet, offering telephone, video, multimedia streaming and data on the same wired and wireless infrastructure is the paramount example of this trend.

<sup>2</sup>This has become a killer application in cellular systems thanks to the widespread use of camera-equipped mobile phones.

in error at the source decoder input produces a very large distortion of the reconstructed source.

The purpose of this paper is to outline a new *pragmatic* approach to the design of JSCC schemes. Instead of focusing on idealized sources, channels and distortion measures as in most information-theoretic literature on JSCC (e.g., see [5] and references therein), we start from the typical structure of state-of the art source coders, as outlined in [6]. Although the focus of this paper is on *practical system design*, our approach is based on two key information theoretic results: a) linear codes achieve the Shannon limit of  $C/\overline{H}$  source symbols per channel use in order to transmit an arbitrary source with *sup-entropy rate*  $\overline{H}$  over a symmetric memoryless channel with capacity  $C$  (see Theorem 1 in the following); b) under quadratic distortion, the concatenation of (dithered) scalar quantization with entropy coding achieves the rate-distortion function of a stationary ergodic source within bounded rate penalty [7, 8].

In our view, result (b) is the theoretical foundation of most state-of the art source “transform” coders, that are based on a linear transformation in order to project the source onto a convenient basis, followed by scalar quantization of the transform coefficients and entropy coding of the quantization indices. The latter is based on a carefully designed probability model matched to the class of sources to be encoded (see [6] and Section 2.1). The catastrophic behavior said above is due to the presence of the entropy coding stage, that is typically implemented by using standard data compression algorithms such as adaptive arithmetic coding or Huffman coding [6, 9, 1]. Fortunately, thanks to result (a), the conventional entropy coding stage can be replaced by a linear non-catastrophic encoder that maps directly the redundant symbol sequence output by the quantizer into a channel codeword (see Section 2.3).

In order to illustrate our ideas, we use as running example the transmission of digital images over a Binary-Input Output-Symmetric (BIOS) channel and use JPEG2000 [9] as our baseline source coder. Since the sequence of quantization indices is not binary in general, we use a successive encoding and “onion-peeling” decoding architecture: the chain rule decomposition of entropy ensure asymptotic optimality of this approach, that has the advantage of making use of binary linear codes, whose design is well understood. As argued in Section 6, our approach can be readily extended to a variety of channels and sources.

As far as low-complexity decoding is concerned, in Section 3 we reduce the probability model defined by JPEG2000 to a Markovian model, that admits a very simple trellis *Factor Graph*. This yields an efficient joint source-channel iterative decoding scheme based on Belief Propagation (BP) [10, 11]. In Section 4 we discuss the design of the linear

coding stage. In this paper we consider the use of punctured Turbo Codes, although other families of linear codes can be easily used instead. Numerical experiments that demonstrate the advantage of the proposed scheme are presented in Section 5.

**Brief literature survey.** JSCC is a very vast subject and covering it is well beyond the scope of this paper. Here, we focus only on classical and recent works directly related to our approach.

The fact that linear codes with syndrome (linear) encoding achieve entropy for discrete sources is well-known (see [12] and references therein). This result was recently extended to arbitrary sources (also non-stationary non-ergodic) in [13]. We make use of this result to prove our Theorem 1, which is indeed a simple corollary. The optimality of linear codes for (almost) lossless fixed-to-fixed length source coding, together with the advances in channel coding that followed the discovery of Turbo Codes [14] and the re-discovery of LDPC codes [11], spurred an impressive amount of work aimed at using these families of codes with low-complexity BP decoding for data compression (see for example [15, 16, 17, 18, 19, 20, 21, 22, 23, 13]). While a channel coding approach is needed for Slepian-Wolf separated coding of correlated sources, it is quite immediate to verify that it is not competitive with standard fixed-to-variable length entropy coding both in terms of complexity and in terms of performance in the standard lossless data compression setting. The only exception is, perhaps, the algorithm devised in [20, 21, 22, 23, 13] based on closed-loop “doping”, that allows the decoder to achieve zero-error by allowing for some small variability in the encoding length. However, when transmitting over a noisy channel closed-loop doping cannot be applied in a straightforward manner since the encoder cannot replicate exactly the decoding process.

On the other hand, it is well-known that linear codes with *linear* encoding are bounded away from the rate-distortion function [24, 25, 26]. Linear codes in the lossy source coding setting have been proposed as a structured way to construct the quantization codebook. However, the encoder must be non-linear, and typically involves a high complexity. Classical results on this topic are [27, 28, 29, 30, 31] and more recent results can be found, for example, in [32, 33, 34].

We would like to stress the fact that our approach is *very different* from the above works: we do make use of linear codes and *linear* encoding. However, linear encoding is applied to the output of a scalar quantizer in order to map the (redundant) quantization indices onto channel symbols. The scheme is not limited by Ancheta’s negative result [24, 26] because quantization is a non-linear operation. However, the scheme is also not limited by the lossless or almost lossless requirement because we are in a lossy setting. Furthermore, we take advantage of the very low complexity of scalar quantization and

linear encoding. In simple terms, we take the best of both non-linear and linear encoding approaches without paying a high price in terms of performance.

Another set of relevant related works deals with JSCC for redundant data, with a BER (Hamming distortion) performance criterion. This has been recently pursued, for example, in [35, 36]. Our use of punctured Turbo Codes for the linear coding stage is largely inspired by [35], with some differences that shall be pointed out in Section 4. At the decoder we do make use of BP iterative decoding taking the source statistics into account. This approach, generally referred to as joint source-channel *decoding*, or source-aided decoding, is treated in a very large number of works, as for example [37, 38, 39, 40, 41, 42, 43]. Several works do not consider eliminating the entropy coding stage in the encoder as done here, but make use of an iterative decoder that exploits some structure of the entropy code (Huffman, as in [41, 43] or arithmetic, as in [40]) in order to mitigate the catastrophic effect of residual bit errors after channel decoding on the entropy code inverse (decompression). Unfortunately, classical entropy coding algorithms do not lend themselves easily to soft-input soft-output BP decoding. Therefore these iterative scheme have typically high complexity and often not so exciting performance.

As a final remark, we would like to mention that several works considered a milder form of JSCC based on the optimization of the error protection (redundancy) of channel coding in order to optimize the end-to-end distortion performance of some standard *embedded* source coding scheme. This idea, which is directly related to the concept of *unequal error protection*, appears for example in [44, 45, 46, 47, 48, 49, 50] and it is briefly discussed in Section 2.2. We would like to mention that our approach does not make use of unequal error protection on the source data, with the exception of the model probability parameters that must be received error-free. This is similar to the *header* high-protection required by standard separated source-channel coding, and involves only a small fraction, vanishing in the limit of large block length, of the overall source length.

## 2 Main ideas

This section is devoted to the main ideas driving the proposed JSCC scheme. The presentation is kept as general as possible. A more detailed presentation of the encoder and decoder design for the specific example of digital images on BIOS channels is provided in Sections 3 and 4, respectively.

## 2.1 A typical source transform coding architecture

We consider a general transform source coding architecture illustrated by the block diagram of Fig. 1 (see [6] and references therein), inspired by JPEG2000 [9].

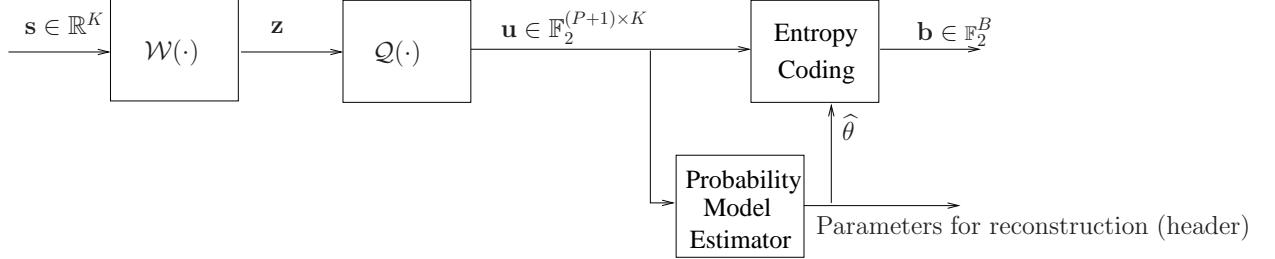


Figure 1: A JPEG2000-like source encoder.  $\hat{\theta}$  denotes the estimated probability model assumed by the entropy coding stage.

The source sequence  $\mathbf{s} \in \mathbb{R}^K$  is transformed by the linear map  $\mathcal{W} : \mathbb{R}^K \rightarrow \mathbb{R}^K$  (e.g., a wavelet transform). The sequence of transform coefficients  $\mathbf{z} = (z_1, \dots, z_K) = \mathcal{W}(\mathbf{s})$  is quantized by applying componentwise the scalar quantizers  $\mathcal{Q}_k : \mathbb{R} \rightarrow \mathbb{F}_2^{P+1}$ , for  $k = 1, \dots, K$ , such that  $u_k = \mathcal{Q}_k(z_k)$ .<sup>3</sup> We denote by  $P$  the number of quantization bits used to represent the magnitude of sample  $z_k$ , and use one additional bit to represent the sign of  $z_k$ . Notice that this binary representation for the quantization indices does not involve any loss of generality. We let  $\mathbf{u} = \mathcal{Q}(\mathbf{z}) = (u_1, \dots, u_K)$  denote the sequence of quantization indices, and let  $\{u_{p,k} : p = 0, \dots, P\}$  denote the “bits” forming  $u_k$ . We think of  $\mathbf{u}$  as a two-dimensional  $(P+1) \times K$  binary array, where the rows  $\mathbf{u}^{(p)} = (u_{p,1}, \dots, u_{p,K})$ , for  $p \in \{0, \dots, P\}$ , are referred to as the  $p$ -th “bit-planes”. The bit-plane  $p = 0$  contains the sign bit, and the bit-planes  $p = 1, \dots, P$  contain the magnitude bits, where  $p = 1$  corresponds to the least significant bit and  $p = P$  corresponds to the most significant bit.

The sequence of quantization indices  $\mathbf{u}$  is generally redundant. Therefore,  $\mathbf{u}$  must be further compressed by a combination of *decimation* and *entropy coding*. By “decimation” we mean discarding some segments of the bit-planes: only the segments of  $\mathbf{u}$  that are not discarded are effectively entropy-encoded and contribute to the encoder output  $\mathbf{b} \in \mathbb{F}_2^B$ . For the sake of exposition simplicity, in this work we assume that no decimation is performed, although the case of decimation can be easily handled.

The entropy encoder is based on a probability parametric model, targeted to the specific class of sources (see [6] and references therein). In the sequel, we assume the

<sup>3</sup>The quantizers  $\mathcal{Q}_k$  may differ by their quantization regions and reconstruction points. For example, in the scheme considered here the quantizers  $\mathcal{Q}_k$  have a different dynamic range depending on which subband of the Discrete Wavelet Transform  $\mathcal{W}$  the symbol  $z_k$  belongs to.

following probability model for our source: let  $\{\mathbb{P}_\theta^{(K)}(\cdot) : \theta \in \Theta, K = 1, 2, \dots\}$  denote a family of processes, defined by the sequence of  $K$  dimensional probability distributions<sup>4</sup> on  $\mathbb{F}_2^{(P+1) \times K}$  parameterized by  $\theta$ . For the time being, we assume that for each  $\theta \in \Theta$ , the corresponding process is a stationary and ergodic, and denote its entropy rate by  $H_\theta(U)$ . This assumption shall be revisited in Section 3 when we discuss more specifically the probability model underlying JPEG2000. Furthermore, we assume that the probability model is *matched*, that is,  $\mathbf{u} \sim \mathbb{P}_\theta^{(K)}$  for some  $\theta \in \Theta$ . A model mismatch would involve additional rate penalty. However, discussing mismatch in this context is rather pointless since the *actual* statistics of real-life sources such as images is essentially unknown.

The modeler estimates the probability parameter  $\hat{\theta}$  from the current realization of  $\mathbf{u}$  and encodes losslessly the  $p$ -th bit-plane using about

$$B_p = -\log_2 \mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u}^{(p)} | \mathbf{u}^{(p+1)}, \dots, \mathbf{u}^{(P)}) \quad (1)$$

bits. As we will see in Section 3, in our working example the probability model is conditionally Markov with fixed state transition diagram, where the conditioning on the  $p$ -th bit-plane is due to the upper bit-planes  $p + 1, \dots, P$ . Then, the model parameter  $\theta$  consists of the collection of transition matrices defining these Markov chains, the elements of which can be easily Maximum-Likelihood (ML) estimated by counting the empirical frequency of symbols corresponding to each state transition.

In source coding schemes such as JPEG2000, the model parameter estimation and the entropy coding is performed simultaneously, by sequential estimation of the state transition probabilities with a Krichevsky-Trofimov (KT) probability estimator [51] together with arithmetic coding [52, 9]. In Fig. 1 we represent the probability model estimation and the entropy coding as two separate blocks for conceptual simplicity and because in the proposed scheme (see Section 2.3) these two functions are indeed performed separately.

The output length of the source encoder is given by  $B = \sum_{p=0}^P B_p$ . It follows that the source coding rate is given by

$$R_s = \frac{B}{K} \text{ bit/source sample}$$

This corresponds to some target distortion level. In the case of no decimation, distortion is due uniquely to the quantization error. We shall refer to this distortion value, denoted by  $D_Q$ , as the *quantization distortion*. Of course, a more flexible tradeoff between distortion and rate can be achieved by considering decimation.

---

<sup>4</sup>In the interest of notation simplicity, in this paper we use the symbol  $\mathbb{P}$  to denote probability with respect to the appropriately defined joint probability space. We do not distinguish between the random variables and the arguments of the probability density or mass function, since it is clear from the context.

## 2.2 Separated approach and Shannon limit

We consider the problem of transmitting the source sequence  $\mathbf{s}$  over a stationary memoryless BIOS channel [11] with capacity  $C$  bit/channel use.

In a classical SSCC approach, the compressed bit sequence  $\mathbf{b}$  produced by the source encoder is mapped into a channel codeword  $\mathbf{x} \in \mathbb{F}_2^N$  by a channel encoder of rate  $R_c = B/N$ . The resulting encoding efficiency  $\eta$  (measured in source samples per channel use) is given by

$$\eta = \frac{K}{N} = \frac{R_c}{R_s} \quad (2)$$

In the limit of large  $K$ , assuming stationarity and ergodicity of the source and a ML probability estimator such that  $\hat{\theta} \rightarrow \theta$ , we have that  $B \rightarrow KH_\theta(U)$ . Furthermore, when  $\Theta$  is an  $M$ -dimensional compact set, Rissanen's bound [53] ensures that the model representation redundancy grows only as  $\frac{M}{2} \log K$ , and therefore communicating the probability model parameter  $\hat{\theta}$  as side information to the decoder costs asymptotically a vanishing rate penalty (more details on the parameter representation are given in Section 3).

From what said above and Shannon source and channel coding theorems [1] it follows that, for arbitrary  $\delta, \epsilon > 0$  and sufficiently large  $K$ , there exist pairs of source and channel codes achieving efficiency

$$\eta = \frac{C}{H_\theta(U)} - \delta \quad (3)$$

with error probability  $\mathbb{P}(\hat{\mathbf{b}} \neq \mathbf{b}) \leq \epsilon$ , where  $\hat{\mathbf{b}}$  denotes the channel-decoded sequence of information bits.

The point with coordinates  $(C/H_\theta(U), D_Q)$  on the efficiency-distortion plane corresponds to the best possible performance for a JSCC with fixed quantizer  $\mathcal{Q}$  and source model parameter  $\theta$ . This point shall be referred to as the *Shannon limit* for our system. Notice that for a source with rate distortion function  $R(D)$ , the best possible efficiency is given by  $C/R(D_Q) \geq C/H_\theta(U)$ . Nevertheless, for complicated sources such as images the rate distortion function is not generally known. Furthermore, scalar quantization followed by entropy coding is near-optimal over a wide range of rates a wide class of sources [54, 7, 8, 55, 56]. Following the pragmatic approach advocated in [6], we say that the rate-distortion point  $(H_\theta(U), D_Q)$  is a point on the source encoder *operational rate-distortion curve*.

As already argued in Section 1, it is well-known that for practical (finite) source block lengths and channel encoding/decoding complexity the SSCC scheme obtained by the concatenation of the transform coder described in Section 2.1 with a channel code might perform quite poorly. It is well-known that entropy coding has a *catastrophic* behavior: its

inverse function is ill-conditioned. A small number of bit-errors in the channel-decoded bit sequence  $\hat{\mathbf{b}}$  generates a large number of symbol errors in the entropy-decoded quantization index sequence  $\hat{\mathbf{u}}$ , and eventually a large distortion in the reconstructed sequence  $\hat{\mathbf{s}} = \mathcal{W}^{-1}(\mathcal{Q}^{-1}(\hat{\mathbf{u}}))$ .<sup>5</sup>

The catastrophic behavior of the source encoder can be partially mitigated by imposing an *embedded* structure. An encoder is said to be embedded if for any  $B' < B$ , the output sequence  $\mathbf{b}'$  produced for output length  $B'$  is the prefix of sequence  $\mathbf{b}$  produced for output length  $B$ . With an embedded source encoder, all the bits in  $\hat{\mathbf{b}}$  from the beginning to the first occurred bit-error can be used for reconstruction, while all the rest must be discarded. Based on this idea, several works (see for example [44, 45, 46, 47, 48, 49, 50]) have addressed the optimization of channel coding redundancy in order to maximize the average number of correctly received bits or minimize the average distortion before the occurrence of the first bit-error. In this paper we take a different approach, outlined in the next section.

### 2.3 The proposed approach

Instead of *mitigating* the catastrophic behavior of the entropy coding stage, we avoid it by taking a JSCC approach: we merge entropy coding and channel coding into a single non-catastrophic encoding operation, that maps the redundant sequence  $\mathbf{u}$  directly into the channel codeword  $\mathbf{x}$  (see Fig. 2).

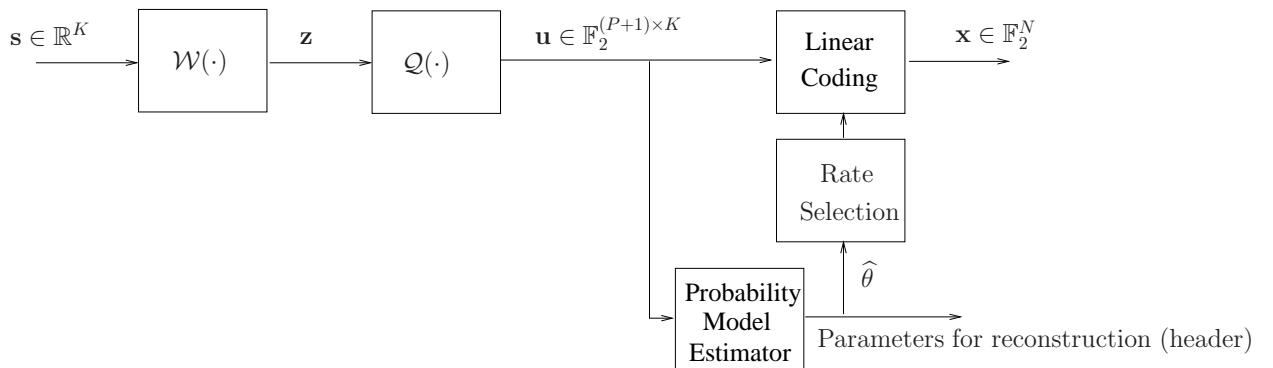


Figure 2: Proposed joint source-channel coding scheme. The estimated probability model parameters  $\hat{\theta}$  are separately transmitted as side information.

Since linear codes achieve the capacity of BIOS channels [57] we shall consider a linear

---

<sup>5</sup>With some abuse of notation, we denote by  $\mathcal{Q}^{-1}$  the reconstruction mapping of the quantizer, i.e.,  $\mathcal{Q}^{-1}(\mathbf{u})$  denotes the representation point  $\hat{\mathbf{s}}$  of the quantization bin indexed by  $\mathbf{u}$ .

map  $\mathbf{u} \mapsto \mathbf{x}$ . Furthermore, since *binary* linear codes are particularly simple and well understood, we shall implement this linear mapping in layers, bit-plane by bit-plane. In particular, we consider  $P + 1$  linear codes  $\mathcal{C}_0, \dots, \mathcal{C}_P$  with block length  $N_0, \dots, N_P$  and generator matrices  $\mathbf{G}_0, \dots, \mathbf{G}_P$ . We obtain the codeword  $\mathbf{x}$  as the concatenation of  $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(P)}$ , where

$$\mathbf{x}^{(p)} = \mathbf{u}^{(p)} \mathbf{G}_p \quad (4)$$

Three questions naturally arise at this point: 1) Suppose that for each  $\theta$  we can pick the best encoding matrices, can we approach the Shannon limit? 2) Is it necessary to “fine tuning” the encoding matrices  $\{\mathbf{G}_p\}$  for each set of source parameters  $\theta \in \Theta$ ? 3) Can we find a low-complexity decoder for the JSCC scheme?

Questions 1 and 2 are addressed simultaneously by Theorem 1 here below and by the comment that follows. Question 3 is addressed in Section 4, where the Markov structure of the source probability model is exploited in conjunction with the structure of the binary linear codes in order to obtain a low-complexity iterative joint source-channel decoder based on BP [10, 11, 14].

The asymptotic goodness of our scheme is supported by the following result, that is an immediate corollary of the optimality of linear codes for both lossless compression [57, 22, 13] and for achieving capacity of BIOS channels (see [57] and references therein) and of the fact that the concatenation of two linear maps is a linear map.

**Theorem 1.** Consider a binary source  $V$  defined by the sequence of  $K$ -dimensional joint probability distributions  $\{P_V^{(K)}(\mathbf{v}) : K = 1, 2, \dots\}$  over  $\mathbb{F}_2^K$ . Define the sup-entropy rate  $\overline{H}(V)$  of  $V$  as [58, 59] the limsup in probability of the sequence of random variables

$$-\frac{1}{K} \log_2 P_V^{(K)}(\mathbf{v})$$

that is, the infimum of all values  $h$  for which

$$\mathbb{P}\left(-\frac{1}{K} \log_2 P_V^{(K)}(\mathbf{v}) \geq h\right) \rightarrow 0 \quad (5)$$

as  $K \rightarrow \infty$ . Consider a system that, for each length  $K$ , maps source sequences  $\mathbf{v}$  into binary codewords  $\mathbf{c} = \mathbf{v}\mathbf{G}$  of length  $N$ , and transmits  $\mathbf{c}$  over a BIOS channel with capacity  $C$ . Let  $\mathbf{y}$  denote the channel output and  $\psi : \mathbf{y} \mapsto \widehat{\mathbf{v}}$  be a suitable decoder.

Then, for any  $\delta, \epsilon > 0$  and sufficiently large  $K$  there exists a  $K \times N$  matrix  $\mathbf{G}$  and a decoder  $\psi$  such that  $\mathbb{P}(\widehat{\mathbf{v}} \neq \mathbf{v}) \leq \epsilon$  and  $K/N \geq C/\overline{H}(V) - \delta$ .

**Proof.** See Appendix A. □

We argue that Theorem 1 has an important consequence for the joint source-channel coding of a family of sources  $\Theta$  and a fixed BIOS channel. In fact, for each value  $H$  there exists *one sequence* of encoding matrices of increasing block length  $K$  and efficiency arbitrarily close to  $C/H$  such that the decoding error probability vanishes *for all* the source statistics with parameters  $\theta \in \Theta$  such that  $H_\theta(U) = H$ . This fact is seen by considering a mixed source [59]  $V$  defined by the family of distributions  $\{\mathbb{P}_\theta : \theta \in \mathcal{H}\}$ , where  $\mathcal{H} \subseteq \Theta$  is the set of parameters for which  $H_\theta(U) = H$  and where  $\theta$  has a uniform prior probability over  $\mathcal{H}$ . It is easily seen that in this case  $\overline{H}(V) = H$ . Therefore, by Theorem 1, there exists a sequence of coding matrices of increasing block length such that  $\mathbb{P}_\theta(\hat{\mathbf{v}} \neq \mathbf{v}) \leq \epsilon$  and  $K/N \geq C/H - \delta$ .

As far as implementation is concerned, this implies that we need only to design one set of coding matrices  $\{\mathbf{G}_p\}$  for each value  $H$  of the source entropy rate. If “tuning” of the codes were needed for each particular  $\theta \in \Theta$ , the scheme would be impractical. In fact, this would require a very large set of codes that do not differ only by their rate, but also by their structure (e.g., by their generator polynomials in the case of TCs, or degree distributions in the case of LDPCs). On the contrary, families of codes with different rate can be obtained by progressive puncturing of a single “mother” code, in a very convenient way for implementation. This is in fact the approach taken in the code design of Section 4: we define a fine quantization grid on the interval  $[0, 1]$  of possible values of the bit-plane empirical entropy rate and design a coding matrix  $\mathbf{G}_p$  for each quantized rate value. Then, when encoding the  $p$ -th bit-plane, we compute the conditional empirical entropy rate

$$\widehat{H}(U_p|U_{p+1}, \dots, U_P) = -\frac{1}{K} \log_2 \mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u}^{(p)}|\mathbf{u}^{(p+1)}, \dots, \mathbf{u}^{(P)}) \quad (6)$$

and choose the corresponding (pre-designed) encoding matrix.

A similar approach is followed in [20, 21, 22, 13] where linear block codes are used for universal lossless compression. Another possibility consists of using *rateless codes* [60, 61, 23], in order to be able to generate an arbitrary amount of coded symbols and therefore a continuum of rates. This approach was followed in the context of universal lossless compression in [23] and it is briefly discussed in Section 6 as a possible extension of the present work.

For an arbitrarily fine rate quantization grid, we let the codeword block length of the  $p$ -th bit-plane be given by

$$N_p = K \left( \frac{\widehat{H}(U_p|U_{p+1}, \dots, U_P)}{C} + \delta_p \right)$$

where  $\delta_p > 0$  is a small rate margin. The achieved coding efficiency is given by

$$\begin{aligned}\eta &= \frac{K}{\sum_{p=0}^P N_p} \\ &= \frac{1}{\sum_{p=0}^P \frac{\widehat{H}(U_p|U_{p+1}, \dots, U_P)}{C} + \delta_p} \\ &= \frac{C}{\widehat{H}(U)} - \delta\end{aligned}\tag{7}$$

where  $\delta$  is a small positive quantity that vanishes as all  $\delta_p \rightarrow 0$  and  $\widehat{H}(U) = -\frac{1}{K} \log_2 \mathbb{P}_{\widehat{\theta}}^{(K)}(\mathbf{u})$  is the empirical entropy rate of  $\mathbf{u}$ . Since  $\widehat{H}(U) \rightarrow H_\theta(U)$ , it follows that for asymptotically large  $K$  the proposed method approaches the operational Shannon limit defined in Section 2.2.

The advantage of JSCC over SSCC becomes clear in the non-asymptotic regime of moderate  $K$  and practical low complexity channel coding and decoding. In fact, the design of non-catastrophic linear encoders is a very well-known and well understood topic in coding theory [62]. In particular, powerful channel coding families such as Turbo Codes (TC) [14], Low-Density Parity-Check (LDPC) codes [11, 63, 64, 65] and Irregular Repeat-Accumulate (IRA) codes [66, 67] can easily achieve post-decoding BER between  $10^{-3}$  and  $10^{-6}$  provided that their rate is below a certain threshold, that is close to  $C/H_\theta(U)$  even for moderate information block length  $K$ . While, as said before, such values of BER would produce very large distortion in the presence of the entropy coding stage, in the proposed system the BER affects directly the bit-plane components  $u_{p,k}$ . It is clear that some bits in error in the bit-planes yield a small output distortion, since the inverse transform  $\mathcal{W}^{-1}$  is linear, unitary or close to unitary, and hence well-conditioned.

As we will see in Section 4, the proposed decoder based on BP [10, 11, 14] computes efficiently an accurate approximation of the symbol-by-symbol posterior probabilities

$$\{\mathbb{P}(u_{p,k}|\mathbf{y}) : p = 0, \dots, P, k = 1, \dots, K\}\tag{8}$$

where  $\mathbf{y}$  is the channel output corresponding to the transmission of  $\mathbf{x}$ .

As far as the source reconstruction is concerned, we let (without loss of generality) the  $k$ -th scalar quantizer map  $u_k = Q_k(z)$  be

$$\begin{aligned}u_{0,k} &= \begin{cases} 0 & z \geq 0 \\ 1 & z < 0 \end{cases} \\ (u_{1,k}, \dots, u_{P,k}) &= \arg \min_{\mathbf{v} \in \mathbb{F}_2^P} \left| \frac{|z|}{\Delta_k} - \sum_{p=1}^P v_p 2^{p-1} \right|\end{aligned}\tag{9}$$

where the parameter  $\Delta_k$  determines the dynamic range of the quantizer for the  $k$ -th transform coefficient. The corresponding reconstruction function is given by

$$Q_k^{-1}(u_k) = (-1)^{u_{0,k}} \frac{\Delta_k}{2} \sum_{p=1}^P u_{p,k} 2^p \quad (10)$$

Given the symbol-by-symbol posterior probabilities (8) produced by the BP decoder, both hard and soft reconstruction are possible. In the first case, symbol-by-symbol hard decisions  $\hat{u}_{p,k} = \arg \max_{v \in \mathbb{F}_2} \mathbb{P}(u_{p,k} = v | \mathbf{y})$  are used in (10) to generate an estimate  $\tilde{z}_k$  of the  $k$ -th transform coefficient. In the second case, the MMSE (conditional mean) estimator  $\tilde{z}_k = \mathbb{E}[z_k | \mathbf{y}]$  is computed. Assuming zero-mean quantization noise statistically independent of the channel output  $\mathbf{y}$ , this takes on the appealing simple form<sup>6</sup>

$$\tilde{z}_k = \frac{\Delta_k}{2} \tanh \left( \frac{\lambda_{0,k}}{2} \right) \sum_{p=1}^P \frac{2^p}{1 + e^{\lambda_{p,k}}} \quad (11)$$

where we define the a posteriori log-likelihood ratio (LLR) for symbol  $u_{p,k}$  as

$$\lambda_{p,k} = \log \frac{P(u_{p,k} = 0 | \mathbf{y})}{P(u_{p,k} = 1 | \mathbf{y})} \quad (12)$$

After producing the sequence  $\tilde{\mathbf{z}} = (\tilde{z}_1, \dots, \tilde{z}_K)$ , the source is reconstructed by applying the inverse linear transform  $\tilde{\mathbf{s}} = \mathcal{W}^{-1}(\tilde{\mathbf{z}})$ . The soft reconstruction approach defined in (11) is sometimes referred to as “soft-bit” reconstruction in the literature [68, 69].

### 3 Probability model, estimation and lossless compression

From this section to the end of the paper we illustrate in more details an implementation example based on JPEG2000. Despite the loss of generality, by developing this example we hope to corroborate the claims made in Section 2.3 and gain in clarity. Generalizations are discussed in Section 6.

In JPEG2000 the linear transform  $\mathcal{W}$  in the block diagram of Fig. 1 is a Discrete Wavelets transform (DWT) [70, 71, 72]. This determines the subband structure shown in Fig. 3. Let us consider a squared gray scale image of dimension  $n \times n$  pixels. After  $D$  stages of DWT, the transform coefficients are partitioned into  $3D + 1$  squared subbands

---

<sup>6</sup>Of course, this is only an approximation if the BP decoder produces approximations of the true symbol-by-symbol a posteriori probabilities (8).

(named  $LL_D$  and  $HL_d$ ,  $LH_d$  and  $HH_d$ , for  $d = 1, \dots, D$ ) of dimensions  $\frac{n}{2^d} \times \frac{n}{2^d}$ , where  $d$  is the decomposition step and where “L” and “H” stand for *Low* and *High* frequency components, respectively.

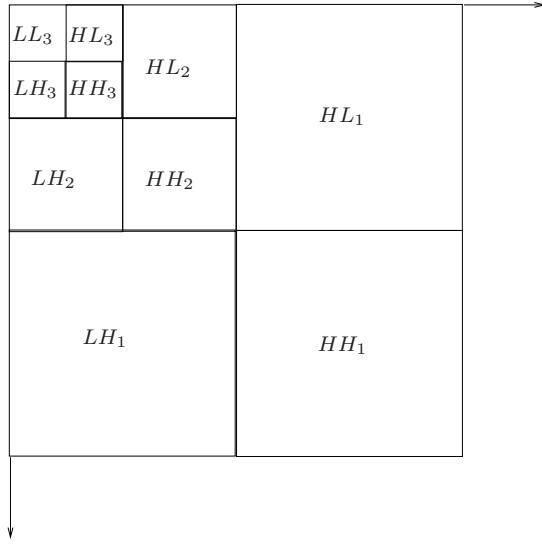


Figure 3: Subband decomposition in the DWT transform with  $D = 3$  levels.

The transform coefficients  $z_k$  are quantized as in (9). The parameter  $\Delta_k$  of the  $k$ -th quantizer is set to a fixed value for all positions  $k$  in the same subband, and depends on the average energy content of the subband [9]. These parameters are sent separately to the decoder for reconstruction. Each quantized subband is partitioned into squared blocks called “code-blocks”, that are independently entropy-encoded. The typical size of code-blocks is  $32 \times 32$  or  $64 \times 64$ . This partitioning is done for the sake of decimation (discarding some code-blocks) and in order to avoid error propagation across code-blocks at the reconstruction side: if an error occurs in the channel-decoded stream, the error propagation will be limited inside a code-block. The probability model is estimated *locally* on each code-block, thus allowing a better matching of the model used for entropy coding with the local statistics of the quantization indices. In JPEG2000 this is obtained by resetting the KT probability estimator at the beginning of each code-block. Since in our system we do not have such error propagation problems and, for simplicity, we do not consider decimation, we shall not use a rigid partitioning into code-blocks. Nevertheless, we shall consider the matching of the probability model parameters to the local statistics, depending on the bit-plane index  $p$  and subband index  $d$ . The binary data corresponding to the bit-plane/subband index pair  $(p, d)$  will be referred to in the following as the  $(p, d)$  “segment”.

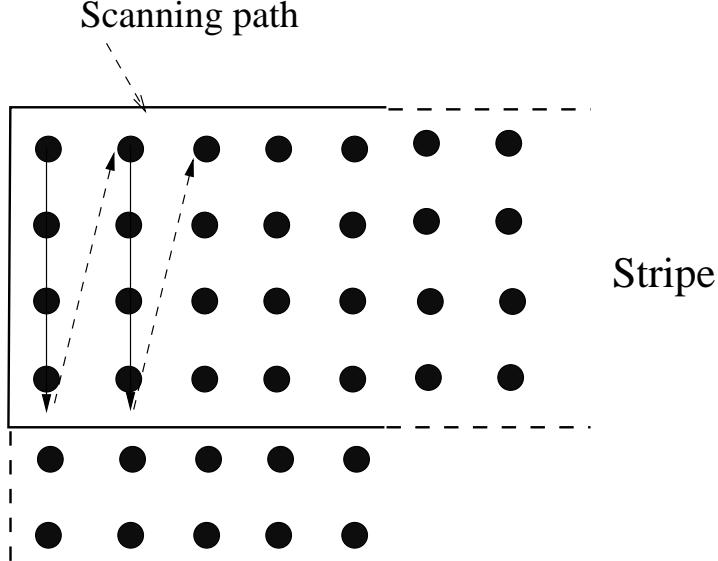


Figure 4: Stripe-oriented scanning of the quantization indices in order to induce a one-dimensional ordering.

The two-dimensional sequence of quantization indices is arranged into a one-dimensional “temporal” ordering according to the so called “stripe-oriented” scanning scheme illustrated in Fig. 4. Each stripe is formed by four rows of quantization indices [9] and a number of columns equal to the dimension of the subband. Then, each quantized coefficient  $u_k$  is identified uniquely by its position  $k$  in the resulting one-dimensional arrangement  $\mathbf{u}$ .

JPEG2000 models each bit-plane  $\mathbf{u}^{(p)}$  as a binary correlated source, where the probability distribution of a bit  $u_{p,k}$  depends on the value of certain neighboring bits in the same plane and on certain bits in the upper bit-planes  $p+1, \dots, P$ . Without entering into the fine details of the scheme, that is extremely tedious and can be learned from standard references [9], we concentrate on the *qualitative* features of the probability model.

The local dependency of bit  $u_{p,k}$  on its neighbors can be formulated as a Markov chain, conditioned on the realization of the symbols in the upper bit-planes  $\mathbf{u}^{(p+1)}, \dots, \mathbf{u}^{(P)}$ . Consider the conditional probability distribution

$$\mathbb{P}_\theta^{(K)}(\mathbf{u}^{(p)} | \mathbf{u}^{(p+1)}, \dots, \mathbf{u}^{(P)}) = \prod_{k=1}^K \mathbb{P}_\theta(u_{p,k} | u_{p,1}, \dots, u_{p,k-1}, \mathbf{u}^{(p+1)}, \dots, \mathbf{u}^{(P)}) \quad (13)$$

The underlying conditional Markov model for the  $p$ -th bit-plane is represented by the block diagram of Fig. 5.

The Markov chain state  $\pi_{p,k} = (u_{p,k-1}, \dots, u_{p,k-L})$  is formed by the content of a *causal* sliding window of previous bits in the same bit-plane (the content of the shift-register of

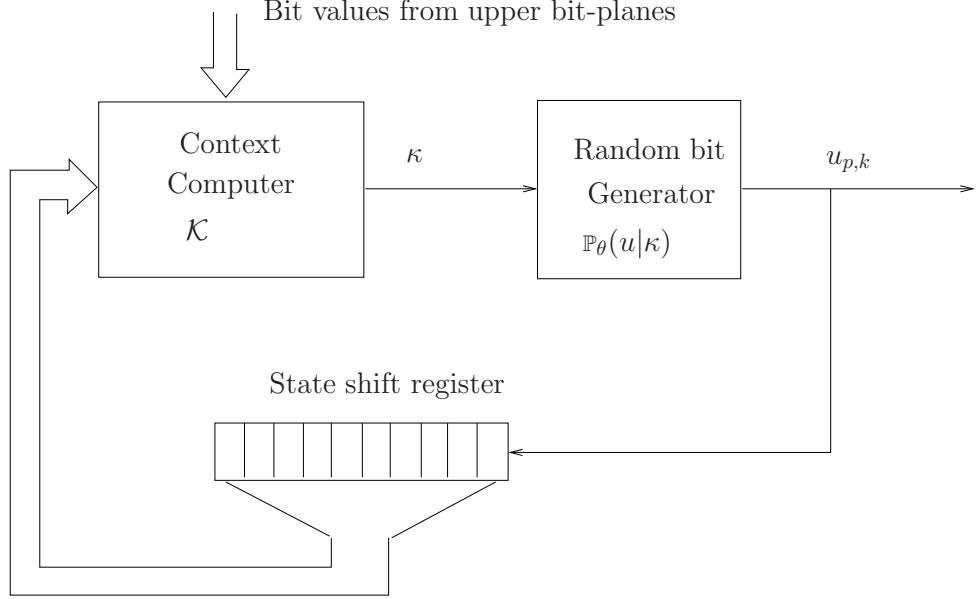


Figure 5: Markov model underlying the conditional bit-plane joint distribution  $\mathbb{P}_\theta^{(K)}(\mathbf{u}^{(p)} | \mathbf{u}^{(p+1)}, \dots, \mathbf{u}^{(P)})$ .

Fig. 5), for some integer  $L$ . Furthermore, the dependency of bit  $u_{p,k}$  on the bits in the upper bit-planes is also confined to a *local* collection of index pairs  $(p', k') \in \mathcal{S}_{p,k}$ . In other words, the local dependency set  $\mathcal{S}_{p,k}$  is defined as the set of index pairs  $(p', k')$  for  $p' > p$  such that

$$\{u_{p'',k''} : (p'', k'') \notin \mathcal{S}_{p,k}, p'' > p\} \rightarrow \{\pi_{p,k}, \{u_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\}\} \rightarrow u_{p,k}$$

is a Markov chain. Obviously, for the top bit-plane we have  $\mathcal{S}_{p,k} = \emptyset$  for all  $k$ .

The Markov model of Fig. 5 has a *fixed* state diagram structure, and it is parameterized by the transition probabilities

$$\mathbb{P}_\theta(u_{p,k} | u_{p,1}, \dots, u_{p,k-1}, \mathbf{u}^{(p+1)}, \dots, \mathbf{u}^{(P)}) = \mathbb{P}_\theta \left( u_{p,k} | \underbrace{u_{p,k-L}, \dots, u_{p,k-1}}_{\pi_{p,k}}, \{u_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\} \right) \quad (14)$$

Fortunately, these probabilities take on distinct values only for a small number of equivalent configurations of the conditioning bits. In the JPEG2000 parlance, we say that the conditioning bits  $(\pi_{p,k}, \{u_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\})$  define the *context* of bit  $u_{p,k}$ . Despite the fact that we have  $2^{L+|\mathcal{S}_{p,k}|}$  possible configurations, many of them are equivalent. We define the context function at bit-plane  $p$  as  $\mathcal{K}_p : \mathbb{F}_2^{L+|\mathcal{S}_{p,k}|} \rightarrow \{0, \dots, M-1\}$ , for some integer  $M$ . We say that two configurations are equivalent if their image under  $\mathcal{K}_p$  (the associated

“context”) is the same. Then, for  $\kappa \in \{0, \dots, M-1\}$  and  $u_{p,k} \in \{0, 1\}$  we have

$$\mathbb{P}_\theta(u_{p,k}|\pi_{p,k}, \{u_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\}) = \mathbb{P}_\theta(u_{p,k}|\kappa)$$

for all configuration of the conditioning bits such that  $\mathcal{K}(\pi_{p,k}, \{u_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\}) = \kappa$ . In JPEG2000 we have  $M = 17$ , where 12 contexts are used for the magnitude bit-planes and 5 contexts are used for the sign bit-plane. More details about the Markov state diagram structure and on context definition in the notation of this paper (which is rather different from the current JPEG2000 descriptions available in the literature [9]) is available in [73]. Fig. 3 shows the trellis diagram corresponding to the Markov chain of Fig. 5 for the top ( $P$ -th) bit-plane. We have  $L = 5$ , which yields a 32-state trellis. The states are enumerated from 0 to 31, the label next to each state contains the state number and the corresponding context  $\kappa$ . There are four non-equivalent trellis sections with the same state transition structure (defined by the shift-register, see footnote (11) in Section 4) but different correspondence between state and context value, that depends on the position of the bit in the stripe. As seen from Fig. 4 there are four different positions corresponding to the four rows forming the stripe. Here, trellis sections from (a) to (d) corresponds to the four positions from top to bottom. Each state has two outgoing transitions corresponding to  $u_{P,k}$  being 0 (solid) or 1 (dashed). For example, for state 26 in section (a), the solid transition corresponds to  $u_{P,k} = 0$  with probability  $\mathbb{P}_\theta(0|5)$  and the dashed transition corresponds to  $u_{P,k} = 1$  with probability  $1 - \mathbb{P}_\theta(0|5)$ . We showed the  $P$ -th bit-plane for the LL and LH subbands. The state-context correspondence for the other subbands is different, and it is determined as explained in [73]. For the  $p$ -th bit-planes with  $p < P$  the correspondence depends on the value of the conditioning bits in the set  $\mathcal{S}_{p,k}$  and then it varies with the time index  $k$ . It would be therefore very cumbersome to represent these trellises in this paper.

At this point, it should be clear that the Markov model is completely determined by the LLRs

$$\nu_{p,k}(\kappa) = \log \frac{\mathbb{P}_\theta(u_{p,k} = 0|\kappa)}{\mathbb{P}_\theta(u_{p,k} = 1|\kappa)}, \quad \kappa = 0, \dots, M-1 \quad (15)$$

In our scheme, we use a non-sequential ML estimator for the above probabilities, that is easily obtained by computing the empirical frequency of zeros for each context  $\kappa$ , where separate bit-counts are maintained for each data block with homogeneous local statistics as explained in the following.

As said before, the Markov model must be matched to the *local* statistics of  $\mathbf{u}$ , that it is in general non-stationary. Different probability estimates are computed for each  $(p, d)$ -th segment. Furthermore, we found that it is convenient to estimate a local statistics for groups of adjacent stripes, generally smaller than a whole segment.

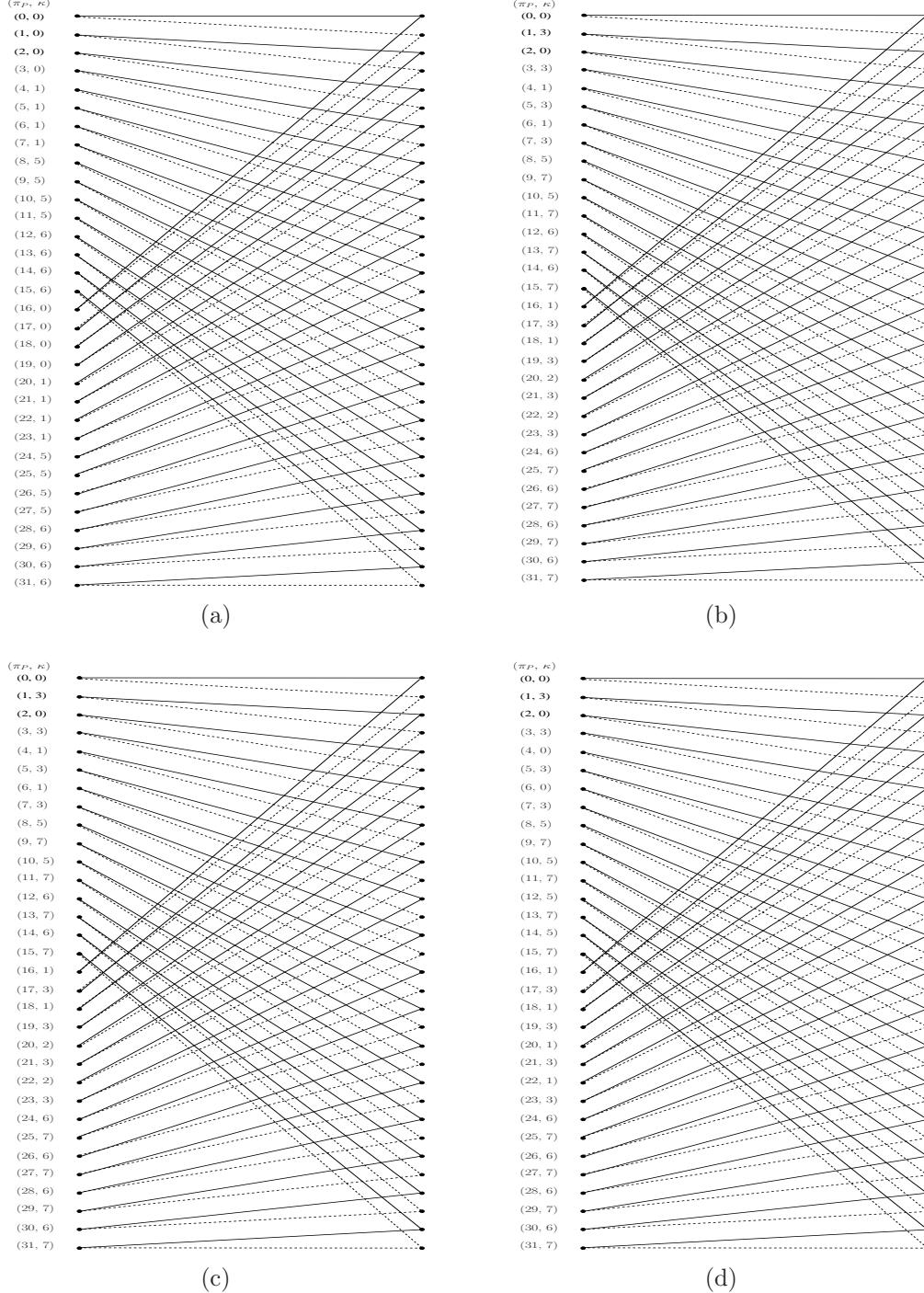


Figure 6: Trellis diagram for the Markov model of the  $P$ -th bit-plane: the Markov chain is time-variant, and the four sections (a), (b), (c) and (d) correspond to the four different positions on a bit in the stripe (see Fig. 4).

Suppose that segment  $(p, d)$  is partitioned into  $m_{p,d}$  regions (groups of adjacent stripes) and over each such region the model transition probabilities are locally estimated. For all positions  $k$  in the same region, the LLRs defined in (15) take on the same value that depend only on the region, and not on the time index  $k$ . We denote such parameters by  $\nu(p, d, \ell, \kappa)$ , where  $p$  denotes the bit-plane,  $d$  denotes the subband,  $\ell = 1, \dots, m_{p,d}$  denotes the region in the  $(p, d)$  segment and  $\kappa$  is the context. Then,  $\nu_{p,k}(\kappa) = \nu(p, d, \ell, \kappa)$  for all positions  $k$  corresponding to the  $\ell$ -th region of the  $(p, d)$ -th segment. It follows that the overall Markov model is piecewise stationary, and it is defined by

$$\mathcal{M} = M_0 \sum_{d=1}^{3D+1} m_{d,0} + M_1 \sum_{d=1}^{3D+1} \sum_{p=1}^P m_{d,p} \quad (16)$$

real parameters, where  $M_1$  denotes the number of distinct contexts for the magnitude bit-planes and  $M_0$  denotes the number of distinct contexts for the sign bit-plane, with  $M_0 + M_1 = M$ . The model parameter  $\theta$  coincides with the collection of all the  $\mathcal{M}$  LLRs  $\{\nu(p, d, \ell, \kappa)\}$  defined above.

The estimated parameters  $\hat{\theta}$  must be sent to the decoder separately and must be highly protected against channel errors, since the decoder needs the probability model for reconstruction (see Section 4). In the next section we shall discuss the compression-only performance of the scheme based on the above defined probability model. We discuss the optimization of the model parameter representation length and, as a sanity check, we compare the compression-only performance obtained by our Markovian probability modeler followed by arithmetic coding with that obtained by the standard JPEG2000.

### 3.1 Compression-only performance

If the probability model and estimator illustrated before is used for compression only, the resulting output length (in bits) is given by  $B_{\text{tot}} = B + B_{\text{model}}$ , where  $B = \sum_{p=0}^P B_p$  with  $B_p$  given in (1) is the number of bits necessary to represent the bit-planes using the estimated probability model (this can be essentially achieved by using arithmetic coding based on the estimated probability  $\mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u})$ ), and where  $B_{\text{model}}$  is the *model redundancy*, i.e., the number of bits necessary to represent the estimated probability model parameter  $\hat{\theta}$ .

Thanks to the energy packing property of the DWT transform, the higher bit-planes are not identically zero only for “low frequency” subbands. We do not encode the identically zero subbands by adding a bit flag in the model parameter to notify the receiver about all-zero subbands.

In order to minimize the total output length,  $B_{\text{model}}$  must be optimized. In particular, we have to choose the number of regions  $m_{p,d}$  to partition each segment, and the number of bits  $q_{p,d}$  for the description of each LLR parameter. To this regard, we have investigated a few possibilities. One option consists of defining regions as groups of  $N_s$  adjacent stripes, where  $N_s$  the same for all segments. Another option consists of having different grouping values  $n_{p,d}$  in each segment  $(p, d)$ . In this case, the values  $n_{p,d}$  for each segment must be added to the model description. As for the quantization bits for parameter representation, one option consists of using Rissanen's description length bound [53]. The achievability part of Theorem 1 in [53] suggests to represent each parameter  $\nu(p, d, \ell, \kappa)$  by  $\lceil \frac{1}{2} \log_2 K_{p,d} \rceil$  bits, where  $K_{p,d}$  denotes the length of the groups in the  $(p, d)$  segment.

Since Rissanen's bound holds on average and it is an asymptotic result, it might not yield the best choice for a given source realization of finite length. Hence, for the sake of comparison, we have also considered a brute-force bit-allocation algorithm to find the global optimum over all values of  $n_{p,d}$  and  $q_{p,d}$ . The brute-force search is initialized by letting  $q_{p,d} = 0$  and  $n_{p,d} = 1$  for all  $p, q$ , which yields equiprobable bits ( $\nu(p, d, \ell, \kappa) = 0$  for all  $p, d, \ell, \kappa$ ). Therefore, the initial value of the total output length is  $B_{\text{tot}} = K(P+1)$ , i.e., the length of the original redundant sequence  $\mathbf{u}$ . Then, we search over all  $n_{p,d} = 1, 2, 3, \dots$  and  $q_{p,d} = 1, 2, \dots$  for the global minimum of the total description length of each  $(p, d)$  segment. The search over  $q_{p,d}$  is stopped when increasing the model parameter quantization bits by one does not correspond to a decrease of the total description length. Even though this search might appear computationally heavy, it should be noticed that in practice only a few values of  $n_{p,d}$  and  $q_{p,d}$  need to be considered. Also, since each segment  $(p, d)$  is independently encoded, the global minimum of the output length is found by independently minimizing the description length of each segment  $(p, d)$ . Hence, the brute-force bit-allocation is actually feasible in practice. In this case, the pair of parameters  $(n_{p,d}, q_{p,d})$  for each segment  $(p, d)$  must be included in the model description is to be added to the total output length.

We run some tests and comparisons based on the monochrome "Goldhill"  $512 \times 512$  and the monochrome "Lena"  $512 \times 512$  test images, after  $D = 2$  stages of Daubechies DWT [70, 9]. Quantization is on 512 levels (corresponding to  $P = 8$ ). In Fig. 7 and 8 we show the values of the total output length as a function of  $N_s$  when the parameter quantization bits are set according to Rissanen's bound. The curves have several local minima and maxima because of integer effects, since the segment lengths are generally not multiples of  $N_s$  and for some values of  $N_s$  we have spurious groups of stripes that cause the fluctuations. It is interesting to notice that the total output length is rather smooth with respect to  $N_s$  and stays close to its global minimum for a wide range of

values. Hence, optimization with respect to  $N_s$  is not very critical in practice, provided that  $N_s$  is chosen reasonably. This suggests that only a few values of  $N_s$  should be tried in a practical implementation.

The values obtained using optimized values  $n_{p,d}$  different for each segment  $(p, d)$  and Rissanen's model bit allocation are reported as a horizontal line, denoted by "Opt. Riss.". Finally, the result of the brute-force bit-allocation is reported as a horizontal line, denoted by "Opt. No. Riss.". In both Fig. 7 and 8 it appears that optimizing with respect to stripe grouping values  $n_{p,d}$  in each segment yields a significant advantage. On the contrary, the parameter quantization bit allocation given by Rissanen's bound is always very close to optimum. Since this allows a much faster optimization of the model description, this method is to be preferred and it is used for the rest of the results presented in this paper.

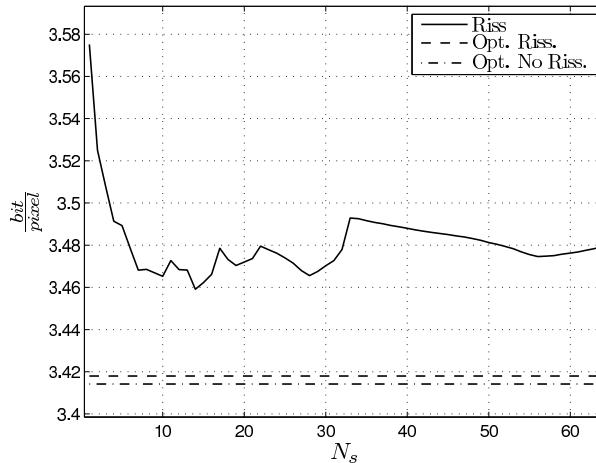


Figure 7: Total output length for various options of the model description for the Goldhill test image.

Table 1 reports (first column) the total output length for "Goldhill" and "Lena" achieved by the proposed probability model using arithmetic coding to compress the data and the above optimized model description. For comparison, the JPEG2000 output length (second column) is reported for the same quantization distortion  $D_Q$ . We notice that there is a remarkable agreement of the overall output length. This shows that the proposed Markov model is consistent with the probability model implicitly assumed by JPEG2000, and that the sequential adaptive entropy coder of JPEG2000 produces a redundancy very similar to the model redundancy obtained by our optimized model description. The slight advantage of our method is believed to be due to the fact that JPEG2000 resets the KT estimators on each code-block and inserts resynchronization

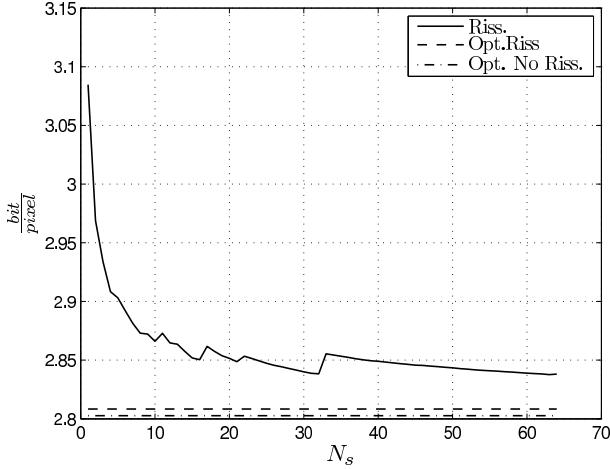


Figure 8: Total output length for various options of the model description for the Lena test image.

Table 1: Comparison between the JPEG2000 output length (in bit per pixel) and the output length of an ideal entropy encoder based on the proposed probability modeler, for  $D = 2$  stages of DWT, 8 bit-planes and optimized number of quantization bits for the model parameters.

Image	Proposed Algorithm	JPEG2000 Encoder
Goldhill	3.45	3.5
Lena	2.81	2.84

symbols to limit error propagation.

## 4 Encoding and decoding

In this section we illustrate the linear joint source-channel encoding and obtain the Factor Graph (FG) of the joint probability distribution of the source sequence  $\mathbf{u}$  and the channel output  $\mathbf{y}$ . The FG yields directly a low-complexity iterative joint source-channel decoder based on BP, by a completely standard application of the Sum-Product computation rules [10]. Since the derivation of the BP algorithm is nothing more than an exercise, it will be omitted for the sake of conciseness.

## 4.1 Linear encoding by punctured Turbo Codes

In order to seek a good tradeoff between performance and complexity, at possibly large but finite block length  $K$ , we resort to the use of well-known families of linear binary codes for which efficient BP decoding is easily implementable. In this paper we focus on the use of TCs [14, 35, 36].<sup>7</sup>

As said in Section 2.3, we successively encode the bit-planes as  $\mathbf{x}^{(p)} = \mathbf{u}^{(p)}\mathbf{G}_p$ , for all  $p = 0, \dots, P$ . We focus on the encoder of a generic  $p$ -th bit-plane and drop the index  $p$  for notation simplicity. We consider a TC family with two *identical* component binary Recursive Convolutional Codes (RCC) of rate 1. The RCC encoder is defined by the input output relation<sup>8</sup>  $x(D) = \frac{b(D)}{a(D)}u(D)$ . As usual, the code generators  $(a(D), b(D))$  are expressed by their coefficients in octal notation. For example, the RCC with generators  $a(D) = 1 + D + D^2 + D^3 + D^4$  and  $b(D) = 1 + D^4$  is indicated by  $(37, 21)_8$ . We use a tail-biting encoder [74, 75, 76]. Hence, the mapping  $(u_1, \dots, u_K) \mapsto (x_1, \dots, x_K)$  is given by

$$\mathbf{x} = \mathbf{u}\mathbf{A}^{-1}\mathbf{B} \quad (17)$$

where  $\mathbf{A}$  is the  $K \times K$  circulant matrix with first row

$$(a_0, a_1, \dots, a_\mu, \underbrace{0, 0, \dots, 0}_{K-\mu-1})$$

and  $\mathbf{B}$  is the circulant matrix with first row

$$(b_0, b_1, \dots, b_\mu, \underbrace{0, 0, \dots, 0}_{K-\mu-1})$$

and  $\mu$  denotes the RCC encoder memory.

The turbo encoder with puncturing is obtained as follows. Let  $\mathbf{\Pi}_1, \mathbf{\Pi}_2$  denote two  $K \times K$  permutation matrices (interleavers), and  $\mathbf{R}_0, \mathbf{R}_1$  and  $\mathbf{R}_2$  denote three *puncturing* matrices, of dimension  $K \times n_0$ ,  $K \times n_1$  and  $K \times n_2$ , respectively. Notice that a  $K \times n$  puncturing matrix is a submatrix of a  $K \times K$  permutation matrix obtained by selecting  $n$  out of  $K$  columns. Then, a generator matrix for the TC with given RCC component, interleaver and puncturing is given by

$$\mathbf{G} = [\mathbf{R}_0 | \mathbf{\Pi}_1 \mathbf{A}^{-1} \mathbf{B} \mathbf{R}_1 | \mathbf{\Pi}_2 \mathbf{A}^{-1} \mathbf{B} \mathbf{R}_2] \quad (18)$$

---

<sup>7</sup>LDPC codes [11, 63, 64, 65] and IRA codes [66, 67], as well as many variations thereof proposed in the recent literature could be used here, after obvious modifications.

<sup>8</sup> $D$ -transform notation: a sequence  $\dots, x_{-1}, x_0, x_1, x_2, \dots$  is represented by the Laurent series  $x(D) = \sum_\ell x_\ell D^\ell$  in the indeterminate  $D$ .

The turbo-encoded output sequence is given by three blocks. The first block (“systematic”),  $\mathbf{u}\mathbf{R}_0$  of length  $n_0$  corresponds to a punctured version of the encoder input. The second and the third blocks (“parity”)  $\mathbf{u}\mathbf{\Pi}_1\mathbf{A}^{-1}\mathbf{B}\mathbf{R}_1$  and  $\mathbf{u}\mathbf{\Pi}_2\mathbf{A}^{-1}\mathbf{B}\mathbf{R}_2$  of length  $n_1$  and  $n_2$ , respectively, are obtained by permuting the input sequence, passing it through the (tail-biting) RCC encoder, and puncturing its output. The coding rate is given by  $K/(n_0 + n_1 + n_2)$ <sup>9</sup>.

Next, we discuss the optimization of the TC encoder parameters, namely, the RCC generator polynomials, interleavers and puncturing matrices. Intuitively,  $\mathbf{G}$  in (18) should *mimic* as closely as possible the generator matrix of a *random* linear code. In particular, a necessary (non-sufficient) condition for approaching the Shannon limit is that the encoder output is *marginally* uniformly distributed [77, 35]. In fact,  $\mathbf{G}$  should map the statistically dependent and marginally non-uniform binary symbols of the input  $\mathbf{u}$  into channel symbols  $\mathbf{x}$  with the required capacity-achieving uniform distribution. This problem is discussed in [35] for the case of non tail-biting TCs and in the limit of infinite block length (a “convolutional coding” framework). In particular, it is shown that the encoder output  $x_k$  marginally uniform when  $k$  is large, for a non-uniform i.i.d. encoder input  $u(D)$  if and only if  $b(D)$  is not a multiple of  $a(D)$ . Also, it is shown that under mild conditions on the generator polynomial  $a(D)$  the state at time  $k$  of the RCC encoders is uniformly distributed over the encoder state space irrespectively of the input probability, for an i.i.d. input sequence and large  $k$ .

Our problem differs from [35] in the fact that we consider a block coding framework and tail-biting codes. Next, we shall show that by choosing a primitive polynomial  $a(D)$ , in the limit of large block length and large polynomial degree we obtain both a marginally uniformly distributed encoder output and an encoder state uniformly distributed over the state space for all positions  $k = 1, \dots, K$  in the block. We have:

**Lemma A.** The mapping  $f(D) \mapsto \mathbf{F}$ , that maps polynomials  $f(D) = f_0 + f_1D + \dots, f_{K-1}D^{K-1}$  in the ring  $[\mathbb{F}_2[D]]_{(1+D^K)}$  of polynomials residues modulo  $1 + D^K$  into the  $K \times K$  circulant matrix whose first row is  $[f_0, f_1, \dots, f_{K-1}]$  is a ring isomorphism.

**Proof.** Clearly, the zero polynomial is mapped into the zero matrix, the polynomial 1 is mapped into the identity matrix, the mapping is linear and preserves the product, i.e.,  $f(D)h(D)$  modulo  $1 + D^K$  (a cyclic convolution of the polynomial coefficients) is mapped into the product  $\mathbf{F}\mathbf{H}$  of the corresponding matrices.  $\square$

As a consequence, we have:

**Lemma B.** A defined above is non-singular if and only if  $a(D)$  is not a divisor of zero

---

<sup>9</sup>This encoder structure cannot implement coding rates smaller than 1/3. If coding rates smaller than 1/3 are needed, we have to add RCC stages in parallel. However, this was not needed in our simulations.

in the ring  $[\mathbb{F}_2[D]]_{(1+D^K)}$ . In particular,  $\mathbf{A}$  is invertible if and only if  $a(D)$  and  $1+D^K$  are relatively primes.  $\square$

Since  $\mathbf{A}^{-1}$  is a circulant matrix and, for what said before, we wish that its rows look as random as possible, we shall choose  $a(D)$  to be a primitive polynomial of degree  $\mu$ . The existence of  $\mathbf{A}^{-1}$  is guaranteed by the following

**Lemma C.** For  $a(D)$  primitive,  $\mathbf{A}$  is invertible if and only if  $2^\mu - 1$  does not divide  $K$ .

**Proof.** For Lemma B, we need that  $a(D)$  and  $1+D^K$  are relatively prime. Since  $a(D)$  is irreducible over  $\mathbb{F}_2$ , it follows that the condition holds if and only if  $a(D)$  does not divide  $1+D^K$ . The roots of  $a(D)$  are primitive elements of the field  $\mathbb{F}_{2^\mu}$ , the extension of degree  $\mu$  over  $\mathbb{F}_2$ . All the non-zero elements of  $\mathbb{F}_{2^\mu}$  are roots of the polynomial  $1+D^{2^\mu-1}$ . Hence,  $a(D)$  is a factor of  $1+D^{2^\mu-1}$ . Finally,  $1+D^{2^\mu-1}$  does not divide  $1+D^K$  if and only if  $2^\mu - 1$  does not divide  $K$ .  $\square$

The condition that  $K$  is not a multiple of  $2^\mu - 1$  is easily satisfied in practice, since typically  $K$  is a power of two. By choosing  $a(D)$  primitive we have that the feedback shift register with coefficients given by  $a(D)$  in the RCC encoder generates an  $m$ -sequence of period  $2^\mu - 1$ , with Hamming weight  $2^{\mu-1}$ . This has the following nice consequence:

**Lemma D.** If  $2^\mu - 1$  does not divide  $K$ , the circulant matrix  $\mathbf{A}^{-1}$  has first row  $\boldsymbol{\tau}$  formed by the concatenation of  $\lfloor \frac{K}{2^\mu-1} \rfloor$  periods of the  $m$ -sequence generated by  $a(D)$ , plus  $K$  modulo  $2^\mu - 1$  extra symbols. In particular, for  $K \gg 2^\mu - 1 \gg 1$  we have that the *normalized* Hamming weight of  $\boldsymbol{\tau}$  is close to  $1/2$ .

**Proof.** Since  $\mathbf{A}^{-1}$  exists, the first row  $\boldsymbol{\tau}$  is obtained by loading the feedback shift register by some non-zero configuration of the memory elements and feeding as input the sequence  $u(D) = 1$ . This clearly generates  $\lfloor \frac{K}{2^\mu-1} \rfloor$  periods of the corresponding  $m$ -sequence, plus some tail symbols to arrive at the total length  $K$ .

The Hamming weight of  $\boldsymbol{\tau}$  is given by

$$\left\lfloor \frac{K}{2^\mu-1} \right\rfloor 2^{\mu-1} + \Delta$$

where  $0 \leq \Delta \leq 2^\mu - 2$  is the Hamming weight of the tail symbols. For large  $\mu$  and  $K \gg 2^\mu - 1$  the above Hamming weight normalized by the block length  $K$  is close to  $\frac{2^{\mu-1}}{2^\mu-1} + \Delta/K \approx 1/2$ .  $\square$

Lemma D implies that we can construct a *structured* generator matrix  $\mathbf{G}$  as in (18) having a marginal empirical distribution of its entries close to  $1/2$ , i.e., close to that of the capacity-achieving linear coding ensemble. For example, consider  $a(D) = 1+D^3+D^4$  (or (23)<sub>8</sub>) and  $K = 16$ . The corresponding first row of  $\mathbf{A}^{-1}$  is equal to

$$\boldsymbol{\tau} = [1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1] \quad (19)$$

and has Hamming weight 9, so that  $9/16 = 0.5625$ . If we consider for example length  $K = 64$ , we would obtain  $\tau$  with Hamming weight 33, so that  $33/64 = 0.5156$ , that is already quite close to  $1/2$ .

By choosing  $b(D)$  of degree  $\leq \mu$  and  $a(D)$  primitive, it follows that  $a(D)$  and  $b(D)$  are relatively prime. From [35] we have that the encoder output  $x_k$  is asymptotically marginally uniformly distributed. However, since we consider a tail-biting code, a cyclic shift of the RCC encoder output corresponds to a cyclic shift of the input and has the same marginal distribution. It follows that, for large block length  $K$ , the output  $x_k$  is marginally uniformly distributed for all positions  $k = 1, \dots, K$ .

Next, we turn our attention to the encoder state. The encoder state space coincides with  $\mathbb{F}_2^\mu$ , which is an Abelian group (with respect to vector addition). In a tail-biting convolutional code, the *circulation state* is defined as the initial encoder state  $\mathbf{s}_0$  such that, for a given input  $u(D)$ , the final state is  $\mathbf{s}_K = \mathbf{s}_0$ , i.e., the path in the trellis corresponding to the initial state  $\mathbf{s}_0$  and input  $u(D)$  closes onto itself (tail-biting condition). The circulation state  $\mathbf{s}$  is a *linear* function of the input  $u(D)$ . In particular, let  $\mathbf{A}'$  denote the  $K \times \mu$  matrix obtained by taking the last  $\mu$  columns of  $\mathbf{A}^{-1}$ . Then, it is straightforward to see that  $\mathbf{s} = \mathbf{u}\mathbf{A}'$ . For example, for the input  $u(D) = 1$  the circulation state corresponding to  $a(D) = 1 + D^3 + D^4$  and  $K = 16$  is given by  $\mathbf{s} = (1, 1, 1, 1)$ , i.e., the last 4 positions of the sequence  $\tau$  in (19). Consider an input sequence  $u(D)$  with i.i.d. symbols, such that  $\mathbb{P}(u_k = 1) = \rho$ . As argued before, a good joint source-channel code should have an encoder state distribution that is uniform over the state space irrespectively of the input bias probability  $\rho$ . This is given by the following:

**Lemma E.** For  $K \gg 2^\mu \gg 1$  and non-uniform i.i.d. encoder input  $u(D)$  with  $\mathbb{P}(u_k = 1) = \rho \in (0, 1)$ ,<sup>10</sup> the circulation state is almost uniformly distributed over the RCC encoder state space. Furthermore, the encoder state at any position in the trellis is also almost uniformly distributed.

**Proof (sketch).** First, we prove the following general simple result.

Fact: Let  $\mathcal{A}$  denote a finite Abelian group of size  $q$ . Consider  $g = \sum_{i=1}^m a_i$  where the elements  $a_i$  are independently selected according to the probability that puts mass  $1/(q-1)$  on all non-zero elements of  $\mathcal{A}$  and probability 0 on the zero element (additive identity of the group). Then, the distribution of  $g$  converges to the uniform distribution on  $\mathcal{A}$  for large  $m$ .

Proof of the Fact: Consider  $m = 2$ , and without loss of generality denote the elements of  $\mathcal{A}$  by the integers  $0, 1, \dots, q-1$ , where 0 denotes the additive identity and the + rule

---

<sup>10</sup>If  $\rho = 0$  or  $\rho = 1$  the entropy of  $u(D)$  is equal to zero, and we shall not encode constant all-zero or all-ones inputs. Hence, this restriction does not involve any loss of generality in our context.

is given by the addition rule of the group. Then,

$$\mathbb{P}(g = i) = \sum_{j=0}^{q-1} \mathbb{P}(i-j)\mathbb{P}(j), \quad i = 0, \dots, q-1 \quad (20)$$

By defining the matrix

$$\mathbf{P} = \frac{1}{q-1} [\mathbf{1}\mathbf{1}^T - \mathbf{I}]$$

(where  $\mathbf{1}$  denotes the all-ones vector of length  $q$ ) and the probability vectors  $\mathbf{p}_0 = (1, 0, \dots, 0)^T$ ,  $\mathbf{p}_1 = (0, 1/(q-1), \dots, 1/(q-1))^T$  and  $\mathbf{p}_2 = (\mathbb{P}(g=0), \dots, \mathbb{P}(g=q-1))^T$  we find that  $\mathbf{p}_1 = \mathbf{P}\mathbf{p}_0$  and that (20) can be written as

$$\mathbf{p}_2 = \mathbf{P}\mathbf{p}_1 = \mathbf{P}^2\mathbf{p}_0$$

Extending this to the case of general  $m$ , we find that the distribution of  $g = \sum_{i=1}^m a_i$  is given by  $\mathbf{p}_m = \mathbf{P}^m\mathbf{p}_0$ . We can write

$$\begin{aligned} \mathbf{P}^m &= \frac{1}{(q-1)^m} \sum_{i=0}^m \binom{m}{i} (-1)^{m-i} (\mathbf{1}\mathbf{1}^T)^i \\ &= \frac{1}{(q-1)^m} \left[ (-1)^m \mathbf{I} + \frac{1}{q} \sum_{i=1}^m \binom{m}{i} (-1)^{m-i} q^i \mathbf{1}\mathbf{1}^T + (-1)^m \frac{1}{q} \mathbf{1}\mathbf{1}^T - (-1)^m \frac{1}{q} \mathbf{1}\mathbf{1}^T \right] \\ &= \frac{1}{(q-1)^m} \left[ \frac{(q-1)^m}{q} \mathbf{1}\mathbf{1}^T + (-1)^m \mathbf{I} - (-1)^m \frac{1}{q} \mathbf{1}\mathbf{1}^T \right] \end{aligned} \quad (21)$$

From (21) it is evident that  $\mathbf{P}^m \rightarrow \frac{1}{q} \mathbf{1}\mathbf{1}^T$  for  $m \rightarrow \infty$ . Then, we conclude that  $\mathbf{p}^m = \mathbf{P}^m\mathbf{p}_0 \rightarrow (1/q)\mathbf{1}$ , as we wanted to show.

In order to show Lemma E, consider a typical realization of the input sequence  $u(D)$ . This has Hamming weight  $m \approx \rho K$ . Furthermore, these can be located in any position of the input sequence with the same probability. The matrix  $\mathbf{A}'$  defined before contains  $\approx K/(2^\mu - 1) \gg 1$  periodic repetitions of the  $2^\mu - 1$  non-zero states plus a segment not longer than  $2^\mu - 2$  states. Each non-zero state appears essentially the same number of times in the rows of  $\mathbf{A}'$ . It follows that a “one” symbol in  $u(D)$  uniformly distributed over all positions  $1, \dots, K$  selects a non-zero state with  $\approx$  uniform probability, equal to  $1/(2^\mu - 1)$ . A typical input selects  $m \approx \rho K$  non-zero elements of the state space with uniform probability. Since the circulation state  $\mathbf{s} = \mathbf{u}\mathbf{A}'$  is given by the sum of these  $m$  random non-zero elements, we apply the above Fact and conclude that, for large  $K$ ,  $\mathbf{s}$  is uniformly distributed over the whole state space  $\mathbb{F}_2^\mu$  (including the zero state).

Furthermore, the RCC encoder state when the encoder is driven by an i.i.d. sequence  $u(D)$  defines a Markov chain. Since  $a(D)$  is primitive, it is easy to see that the Markov

chain is indecomposable and aperiodic. By construction, the transition matrix of the Markov chain corresponding to the RCC state has exactly two non-zero elements in each row and column, equal to  $\rho$  and  $1 - \rho$ . It follows that the (unique) stationary distribution is given by the uniform distribution. Since the circulation state is (almost) uniformly distributed, any state at any given section of the trellis is also (almost) uniformly distributed, and it is exactly uniformly distributed as  $K \rightarrow \infty$ .  $\square$

Notice that the input sequence to the RCC encoder in the tail-biting turbo encoder defined before is an interleaved version of the bit-plane bit sequence. Hence, even if the latter is not i.i.d. (e.g., in our case the bit-plane is a non-stationary Markov chain), after interleaving it will be close to i.i.d., with a probability  $\hat{\rho}$  equal to the fraction of “ones” in the bit-plane. Notice also that, although we interleave the bit-plane sequence for the purpose of turbo encoding and BP decoding, the bit-plane entropy rate is computed taking into account the Markov memory structure, as said in Section 3. This has nothing to do with the entropy rate of the bit-plane *after* interleaving, given by  $H_2(\hat{\rho})$  where  $H_2(\cdot)$  denotes the binary entropy function. By the concavity of entropy the latter may be much larger than the actual bit-plane entropy rate.

For the choice of  $b(D)$  we follow the theory developed in [35]. In particular, letting  $b(D) = 1$  yields that the RCC encoder  $\mu$ -th order empirical distribution, as defined in [77], is uniformly distributed over  $\mathbb{F}_2^\mu$ . This follows immediately from the fact that in this case the encoder output  $x(D)$  coincides with the state bits and for Lemma E we have that the empirical distribution of any block  $(x_{k-\mu}, \dots, x_{k-1})$  state bits is uniformly distributed over  $\mathbb{F}_2^\mu$ . However,  $b(D) = 1$  does not necessarily yield the best coding performance. Hence, we have searched semi-exhaustively for generator polynomials  $b(D)$  with  $b_0 = b_\mu = 1$ .

For given RCC generators, the permutations  $\boldsymbol{\Pi}_1$  and  $\boldsymbol{\Pi}_2$  are chosen at random, by trial and error. For  $K$  not too small, the effect of optimizing the permutations on the end-to-end distortion of the proposed scheme is minimal. In fact, one significant advantage of the proposed JSCC scheme is that its performance is not dominated by the “error floor” region of the BER [14]: as soon as the BER drops to small values (waterfall region) the reconstruction quality of our JSCC scheme rapidly improves and attains satisfactory distortion levels even though the residual BER is not as small as it would be required by (lossless) data applications. In this respect, the proposed JSCC scheme puts much less stress on the code design than a conventional SSCC scheme!

More care must be deserved to the optimization of the puncturing matrices. When the bits in a given bit-plane are not i.i.d. uniformly distributed, sending the systematic part over the BIOS channel results in a suboptimal code. Hence, unless  $\widehat{H}(U_p|U_{p+1}, \dots, U_P)$  is very close to 1 bit/symbol, we let  $n_0 = 0$ . For  $\mathbf{R}_1$  and  $\mathbf{R}_2$ , we have constructed (off-

line) a library of puncturing matrices by following the incremental redundancy approach advocated in [18, 19, 23, 13]. We stress the fact that the codes are constructed off-line, and do not depend on the actual source sequence to be encoded. By using a greedy pseudorandom progressive growth search (i.e., adding columns to the puncturing matrices), we have designed a library of nested puncturing matrices in order to cover all the rates  $C/H - \delta$ , for all quantized values  $H$  in pre-determined fine grid in  $[0, 1]$ . The library was designed assuming a Bernoulli source with entropy  $H$  for all the quantized values of  $H$ . Further discussion on the design of the puncturing matrices is provided in [73] (see also [36]).

Fig. 9 shows the threshold effect for the case of a binary Bernoulli source of length  $K = 512 \times 512$ , with entropy rate  $H(U) = 0.5$ , and a Binary Symmetric Channel (BSC) with capacity  $C = 0.5$ . Notice that each point of Fig. 9 corresponds to a different puncturing matrix, generated by incremental redundancy as explained above. The BER is averaged over a random choice of the interleavers. The transition of the family of progressively punctured TCs generated in this way is remarkably close to the Shannon limit  $\eta = \frac{C}{H(U)} = 1$ . Again, we stress the fact that our scheme does not need very small BER in order to provide good reproduction results. BERs of the order to  $10^{-2}$  are sufficient, as we will see in the examples of Section 5.

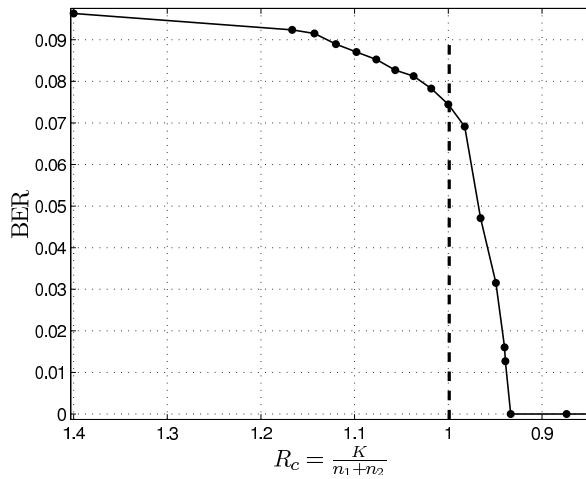


Figure 9: Threshold effect for the case of a Bernoulli source  $H(U) = 0.5$ , transmitted over a BSC  $C = 0.5$ , for a family of progressively punctured TCs.

## 4.2 Belief Propagation decoding

The decoder is based on the successive decoding structure of Fig. 10, where the bit-planes are decoded in sequence, and the decoder at level  $p$  makes use of the hard decisions made at the upper levels. These hard decisions determine the value of the conditioning bits needed for the conditional Markov model at each level. An iterative version of the multistage decoder where soft information is exchanged between the bit-plane decoders was also considered and simulated, but we found that it does not provide any substantial improvements and therefore the additional (significant) complexity is not justified.

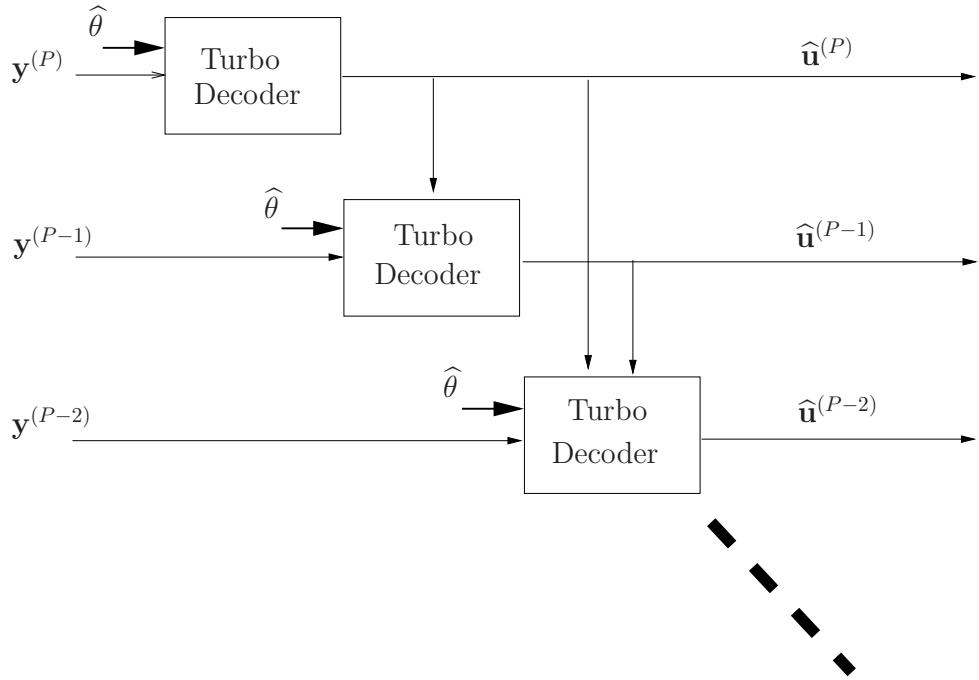


Figure 10: Structure of the successive bit-plane joint source-channel decoder.

It is assumed that the decoder knows the probability model  $\mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u})$  defined in Section 3 with the LLRs given in (15) as model parameters. As already said, the model parameters have to be sent separately, and need to be decoded with very high reliability. Fortunately, the model redundancy is very small in comparison with the data length and therefore we can afford to use a low-rate code to protect the model parameters against channel errors. This is just a standard channel coding problem that we do not address here. However, we would like to stress that the simulation results and the comparison with conventional SSCC is made by *including* the transmission of the model parameters, both in terms of redundancy and in terms of errors, i.e., the comparison is *fair* and was made without any optimistic assumption on perfect reception of the side information  $\hat{\theta}$ .

Let's focus on the  $p$ -th component decoder and let  $\mathbf{y}^{(p)}$  denote the BIOS channel output corresponding to the transmission of the coded  $p$ -th bit-plane  $\mathbf{x}^{(p)} = \mathbf{u}^{(p)}\mathbf{G}_p$ . The goal of the  $p$ -th decoder is to compute the symbol-by-symbol posterior marginal probability

$$\mathbb{P}_{\hat{\theta}}(u_{p,k} | \mathbf{y}^{(p)}, \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}) \quad (22)$$

where  $\hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}$  denote the hard decisions on the upper bit-planes made at previous decoding stages. This can be accomplished (approximately) by the BP algorithm applied to the FG of the underlying joint probability distribution. In the case of Markovian sources and TCs, the BP takes on the particularly appealing form of three BCJR forward-backward algorithms [78] exchanging soft information in the form of “extrinsic” log-likelihood ratios (LLRs) [10]. This follows by standard application of the Sum-Product computation rules [10] to the resulting FG. Hence, this section is devoted to illuminating the FG structure.

Recall the “marginalization” notation of [10] for which  $\sum_{\sim v}$  denotes the sum over all variables in the summation argument while keeping  $v$  fixed. Furthermore, a probability distribution  $\mathbb{P}(v)$  needs to be determined only up to a proportionality constant (denoted by  $\propto$ ), that can be obtained by imposing the normalization  $\sum_v \mathbb{P}(v) = 1$ . Using Bayes rule and neglecting irrelevant proportionality terms, we obtained the factorization

$$\begin{aligned} \mathbb{P}_{\hat{\theta}}(u_{p,k} | \mathbf{y}^{(p)}, \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}) &= \sum_{\sim u_{p,k}} \mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u}^{(p)} | \mathbf{y}^{(p)}, \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}) \\ &\propto \sum_{\sim u_{p,k}} \mathbb{P}(\mathbf{y}^{(p)}, \mathbf{u}^{(p)} | \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}, \hat{\theta}) \\ &= \sum_{\sim u_{p,k}} \sum_{\mathbf{x}^{(p)}} \mathbb{P}(\mathbf{y}^{(p)}, \mathbf{x}^{(p)}, \mathbf{u}^{(p)} | \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}, \hat{\theta}) \\ &= \sum_{\sim u_{p,k}} \sum_{\mathbf{x}^{(p)}} \mathbb{P}(\mathbf{y}^{(p)} | \mathbf{x}^{(p)}) \mathbb{P}(\mathbf{x}^{(p)} | \mathbf{u}^{(p)}) \mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u}^{(p)} | \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}) \\ &= \sum_{\sim u_{p,k}} \sum_{\mathbf{x}^{(p)}} \left[ \prod_{j=1}^{N_p} \mathbb{P}(y_{p,j} | x_{p,j}) \right] \mathbf{1}\{\mathbf{x}^{(p)} = \mathbf{u}^{(p)}\mathbf{G}_p\} \cdot \\ &\quad \cdot \mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u}^{(p)} | \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}) \end{aligned} \quad (23)$$

where we have used the fact that the BIOS channel is memoryless and where  $\mathbb{P}(\mathbf{x}^{(p)} | \mathbf{u}^{(p)}) = \mathbf{1}\{\mathbf{x}^{(p)} = \mathbf{u}^{(p)}\mathbf{G}_p\}$  (an indicator function) since  $\mathbf{x}^{(p)}$  is a deterministic function of  $\mathbf{u}^{(p)}$ .

Eventually, the desired posterior symbol-by-symbol probabilities are obtained by marginalizing the joint distribution in the last line of (23). From (13) and (14), the source proba-

bility term factors as

$$\mathbb{P}_{\hat{\theta}}^{(K)}(\mathbf{u}^{(p)} | \hat{\mathbf{u}}^{(p+1)}, \dots, \hat{\mathbf{u}}^{(P)}) = \mathbb{P}(\pi_{p,1}) \prod_{k=1}^K \mathbb{P}_{\hat{\theta}}(\pi_{p,k+1} | \pi_{p,k}, \{\hat{u}_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\}) \quad (24)$$

where the a priori probability of  $\pi_{p,1}$  is given by

$$\mathbb{P}(\pi_{p,1}) = \begin{cases} 1 & \pi_{p,1} = 0 \\ 0 & \text{otherwise} \end{cases}$$

and where the state transition probability is given by<sup>11</sup>

$$\mathbb{P}_{\hat{\theta}}(\pi_{p,k+1} | \pi_{p,k}, \{\hat{u}_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\}) = \begin{cases} \mathbb{P}_{\hat{\theta}}(u_{p,k} = u | \kappa) & \text{if } \pi_{p,k+1} = (u, \vec{\pi}_{p,k}) \text{ and} \\ & \mathcal{K}_p(\pi_{p,k}, \{\hat{u}_{p',k'} : (p', k') \in \mathcal{S}_{p,k}\}) = \kappa \\ 0 & \text{otherwise} \end{cases}$$

The corresponding FG takes on the form of a trellis [10], reflecting the state transition diagram of the Markov probability model.

The other non-trivial component of the overall FG is given by the factorization of the code indicator function  $1\{\mathbf{x}^{(p)} = \mathbf{u}^{(p)}\mathbf{G}_p\}$ . It is well-known that for TCs this takes the form of two trellises, one for each RCC component, interconnected by interleavers.

The factor graph (FG) corresponding to the factorization in the last line of (23) is given in Fig. 11. We use Wiberg's notation (see [10]), for which the FG is a bipartite graph with variable nodes (circles) and function nodes (boxes). State nodes are denoted by filled circles. A variable node is connected to a function node if the corresponding variable is an argument of the corresponding factor [10]. In our case, the variable nodes correspond to the bit-plane bits  $u_{p,k}$ , to the Markov source states  $\pi_{p,k}$  and to the RCC encoder states. The function nodes correspond to the state transition probabilities and to the BIOS channel transition probabilities. The channel output symbols  $y_{p,j}$  and the conditioning bits from the upper bit-planes appear as “dongles”. The channel outputs corresponding to punctured coded symbols are represented as crossed dongles. These symbols are treated as *erasures* by the decoder.

Standard application of Sum-Product computation rules to this FG yields the iterative BP decoder, where one BCJR algorithm for each trellis subgraph in the FG is used. The three BCJR component decoders exchange messages in the form of LLRs [10] via

---

<sup>11</sup>The update of the shift register in the block diagram of Fig. 5 is given by

$$\pi_{p,k+1} = (u_{p,k}, \vec{\pi}_{p,k})$$

where  $\vec{\pi}_{p,k}$  denotes a right shift by one position of the state register.

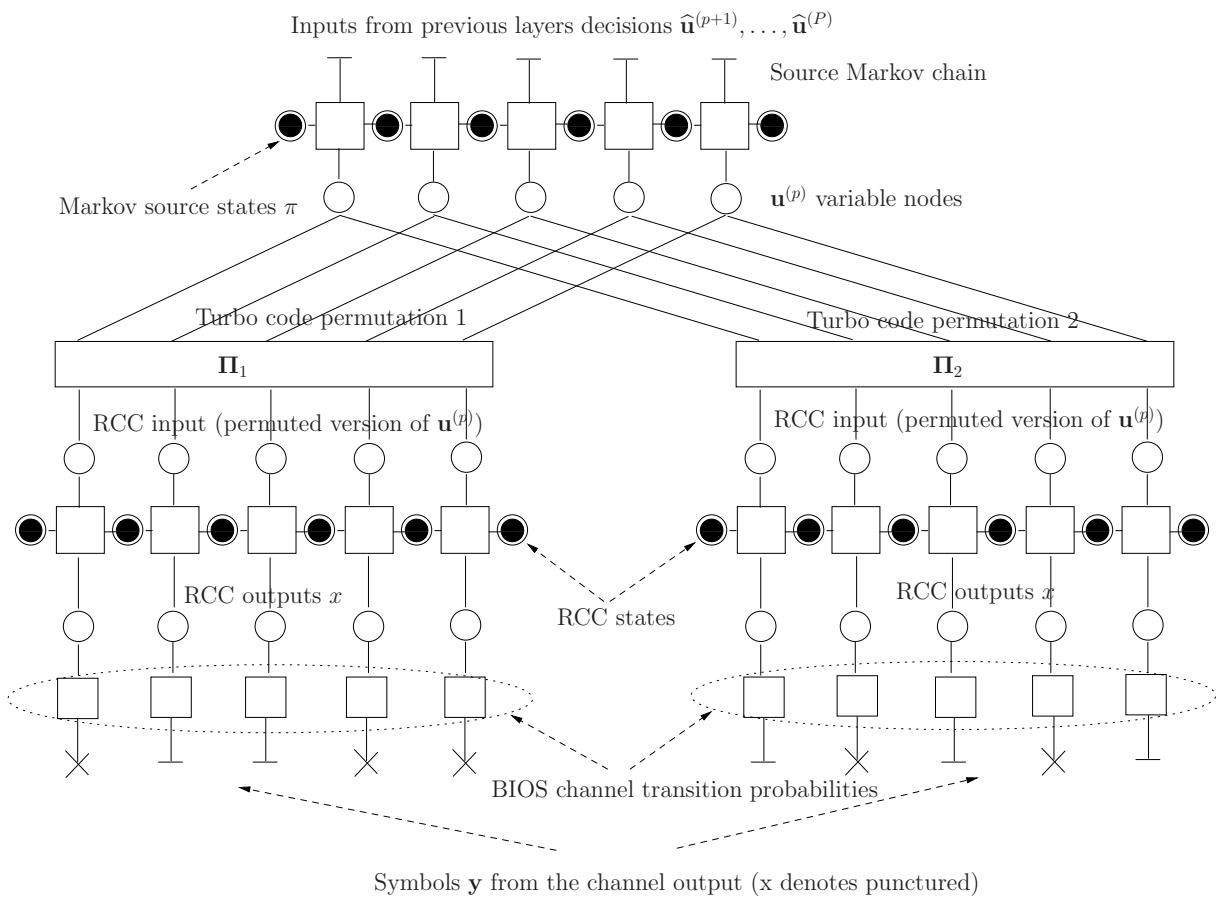


Figure 11: Factor graph underlying the  $p$ -th component BP decoder.

the degree-3 variable nodes  $u_{p,1}, \dots, u_{p,K}$ . In our simulations, we used a BP scheduling where the BCJRs are activated in round-robin (order is basically irrelevant). It should be noticed that symbol-by-symbol hard decisions are made for each stage  $p$  and are used in subsequent stages as a priori information. However, in the soft-reconstruction scheme, the symbol-by-symbol a posteriori probabilities for each bit-plane are used in order to compute the reconstructed source via (11).

## 5 Results

In this section we present some results in terms of Peak Signal to Noise Ratio (PSNR)<sup>12</sup> versus efficiency  $\eta$ , measured as the number of source samples (pixels) per channel use. We considered two 8-bit gray scale test images of size  $512 \times 512$  pixels, referred to as “Goldhill” and “Lena”. We run simulations over a Binary Symmetric Channel (BSC) and we designed our schemes for a BSC cross-over probability  $\rho = 0.05$  (corresponding to  $C = 1 - H_2(\rho) = 0.7136$  bits per channel use).

For comparison, we have considered a SSCC scheme obtained by concatenating the output of the JPEG2000 encoder with a standard punctured TC with rate optimized in order to work as close as possible to the channel capacity. The simulation were performed by using the OpenJPEG library, an open-source JPEG2000 codec written in C language [79]. We consider the case in which the error resilience tools provided by JPEG2000 standard (i.e SOP marker, SEGMARK marker and ERTEM strategy [9]) are enabled. In order to compute the performances of the JPEG2000 compressed images, we transmit and protect separately the header and the markers: they are received error free by the decoder.

As for the SSCC, several generator polynomials have been tested. Here we report only the case of  $(37, 21)_8$  [14], that yields the best results. Since the output of JPEG2000 can be considered as i.i.d. uniform bits, the TC are conventional and we used standard regular puncturing patterns available in the literature.

We would like to stress the fact that these results are presented here for the purpose of a *proof of concept*: no particular effort has been made to optimize the codes beyond the simple incremental redundancy greedy selection of the puncturing patters described before. We hasten to say that the result might be probably improved by better code

---

<sup>12</sup> $PSNR \triangleq 10\log_{10} \frac{A^2}{MSE}$  is a measure of quality of image reconstruction where  $MSE = \frac{1}{K}\mathbb{E}[|\mathbf{s} - \tilde{\mathbf{s}}|^2]$  is the Mean Squared distortion of the reconstructed source and  $A$  is the peak-to-peak signal amplitude. In our case,  $A = 255$ , since we are working with 8-bit images. We fix the source realization to be one of the test images considered and we take expectation  $\mathbb{E}[\cdot]$  with respect to the channel noise.

design. Furthermore, as mentioned in the literature review of Section 1, a whole range of schemes between the basic SSCC scheme considered here and a fully JS SCC are available, based on tuning the channel coding redundancy for each JPEG2000 encoded block [44, 45, 46, 47, 48, 49, 50]. We did not consider comparisons with these schemes since they are *considerably* more complicated than the proposed JS SCC, and they are actually *not yet used* in today’s applications. Given the number and the variety of schemes proposed in the literature, a full comparison is well beyond the scope of this paper.

Tables 2 and 3 summarize the main characteristics of the JS SCC schemes for the two test images considered, respectively. The number of encoded bits in each bit-plane is indicated by  $K_p$ . This may be less than  $512^2 = 262144$  since some segments, especially for the top bit-planes, might be identically zero. The third column shows the empirical entropy rate  $\widehat{H}_p \triangleq \widehat{H}(U_p|U_{p+1}, \dots, U_P)$  as defined in (6). The fourth column shows the bound on coding efficiency for each bit-plane given by  $\bar{\eta}_p = C/\widehat{H}_p$ . We used generators  $(23, 35)_8$  for all bit-planes.

Table 2: Goldhill

Bit-plane $p$	length $K_p$	empirical entropy $\widehat{H}_p$	efficiency $\bar{\eta}_p$
8	16384	0.2966	2.4063
7	29184	0.3158	2.2595
6	71680	0.2468	2.8906
5	142336	0.2341	3.0474
4	223232	0.3124	2.2837
3	253440	0.5431	1.3138
2	262144	0.8172	0.8731
1	262144	0.9586	0.7444
0 (Sign)	262144	0.6138	1.1626

In order to deliver the model parameters to the decoder with high reliability we have separately encoded them using a conventional TC (generators  $(37, 21)_8$ , rate  $1/3$ ), that achieves practically error-free performance for the BSC channel with capacity 0.7136. For reconstruction we have used the soft-bit decoding strategy given at the end of Section 2.3.

Figs. 12 and 13 show the PSNR performance of the proposed JS SCC scheme and the conventional SSCC. The resulting PSNR is plotted versus the coding efficiency, in order to put in evidence the gap of the actual coding scheme with respect to the Shannon limit defined as in Section 2.2. Notice that the coding efficiency decreases while moving

Table 3: Lena

Bit-plane $p$	length $K_p$	empirical entropy $\bar{H}_p$	efficiency $\bar{\eta}_p$
8	16384	0.345	2.0691
7	29184	0.258	2.7598
6	71680	0.277	2.5764
5	142336	0.191	3.7315
4	223232	0.190	3.7527
3	253440	0.324	2.1972
2	262144	0.672	1.0607
1	262144	0.925	0.7713
0 (Sign)	262144	0.4977	1.4337

towards the right of the horizontal axis. As the coding efficiency decreases, the PSNR reaches its maximum value which corresponds to the quantization distortion  $D_Q$ , equal to  $PSNR_{Goldhill} = 49.57$  and  $PSNR_{Lena} = 49.08$  dB for “Goldhill” and “Lena”, respectively.

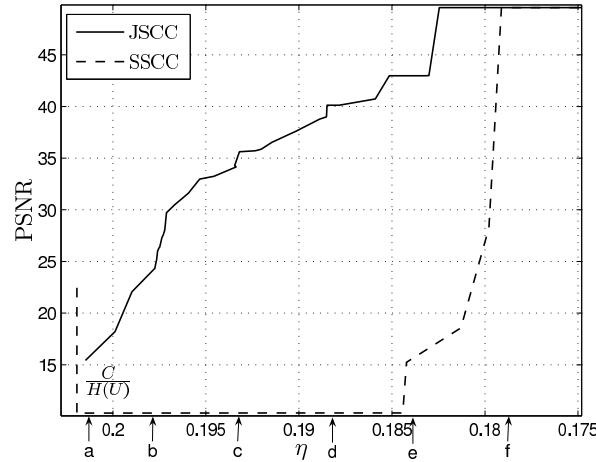


Figure 12: PSNR comparison between SSCC schemes and JSSC schemes for “Goldhill”.

PSNR curves may not tell much about the actual quality of the reconstructed image. Hence, we include also *snapshots* of the reconstructed images for both the JSSC and the SSCC schemes corresponding to the efficiencies marked on Figs. 12 and 13 as “a,b,c,d,e,f”. The corresponding values  $\eta_a, \eta_b, \dots, \eta_f$  have increasing gap from the Shannon limit.

These snapshots, shown in Figs. 14 and 15, illustrate the main claims of this paper and provide experimental evidence about the effectiveness of the proposed JSSC scheme:

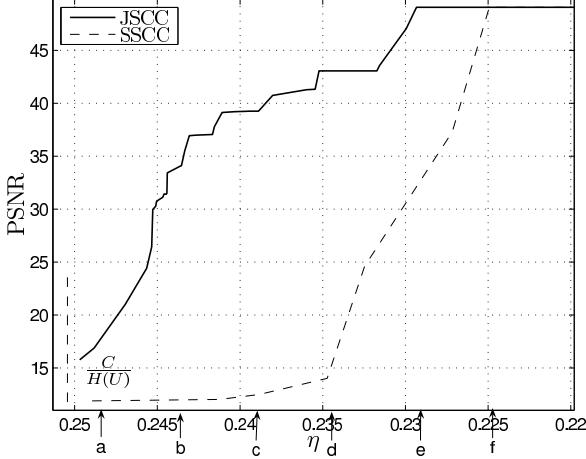


Figure 13: PSNR comparison between SSCC schemes and JSSC schemes for “Lena”.

it is clearly visible that as soon as  $\eta$  is slightly above the Shannon limit, our JSSC scheme achieves an acceptable reconstruction quality. On the contrary, the conventional SSCC achieves very poor quality for a much wider range of the gap from Shannon limit, and suddenly achieves the quantization distortion when the channel code is able to eliminate completely the channel errors (with very high probability). This behavior reflects the catastrophicity of the entropy coding stage of the conventional scheme, which is essentially *eliminated* by our approach. Moreover, in the SSCC the reconstructed images clearly show the square patterns due to the catastrophic error propagation inside code-blocks, because of residual bit errors after channel decoding. Even when the PSNR is not so low, these patterns reduce significantly the perceived image quality. On the other hand, the reconstructed images generated by the proposed JSSC method present a “salt and pepper” noise, that is less annoying for the perceived image quality. This nature of the generated noise opens the door to the study of the concatenation of our approach with the recently proposed discrete universal denoising algorithm (DUDE) [80], that can take advantage of the apparently independent memoryless residual noise after JSSC decoding in order to further improve the image visual quality. This interesting approach is not pursued in this paper and it is mentioned in Section 6 among the directions for future work.

## 6 Conclusion

We presented a new scheme for pragmatic Joint Source-Channel Coding that builds over two information theoretic key ideas: a) the optimal rate-distortion performance can be

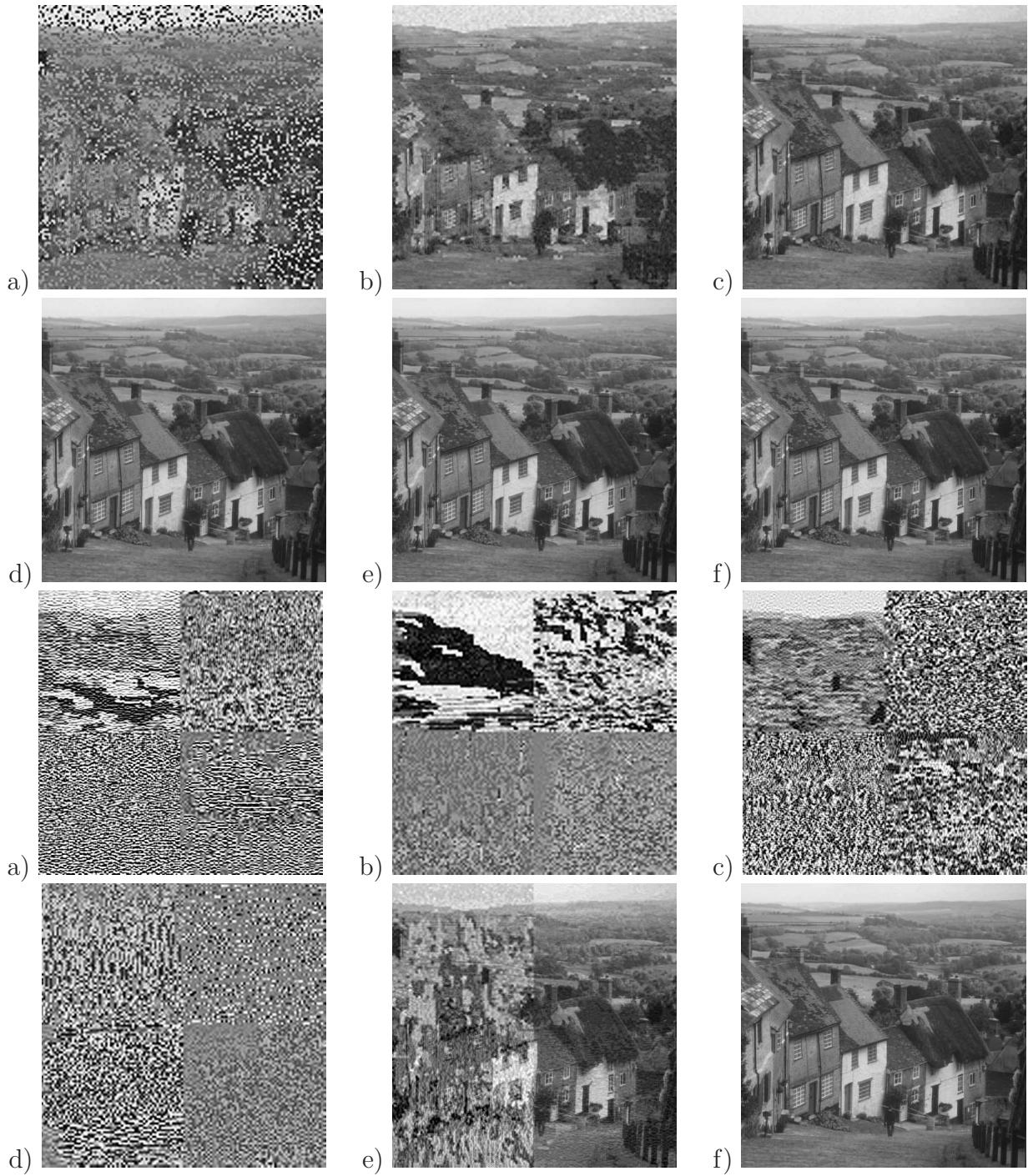


Figure 14: Visual comparison for JSSC (first set) and the SSCC (second set) and the Goldhill image, for decreasing values of  $\eta_a, \dots, \eta_f$  indicated in Fig. 12 (from left to right).

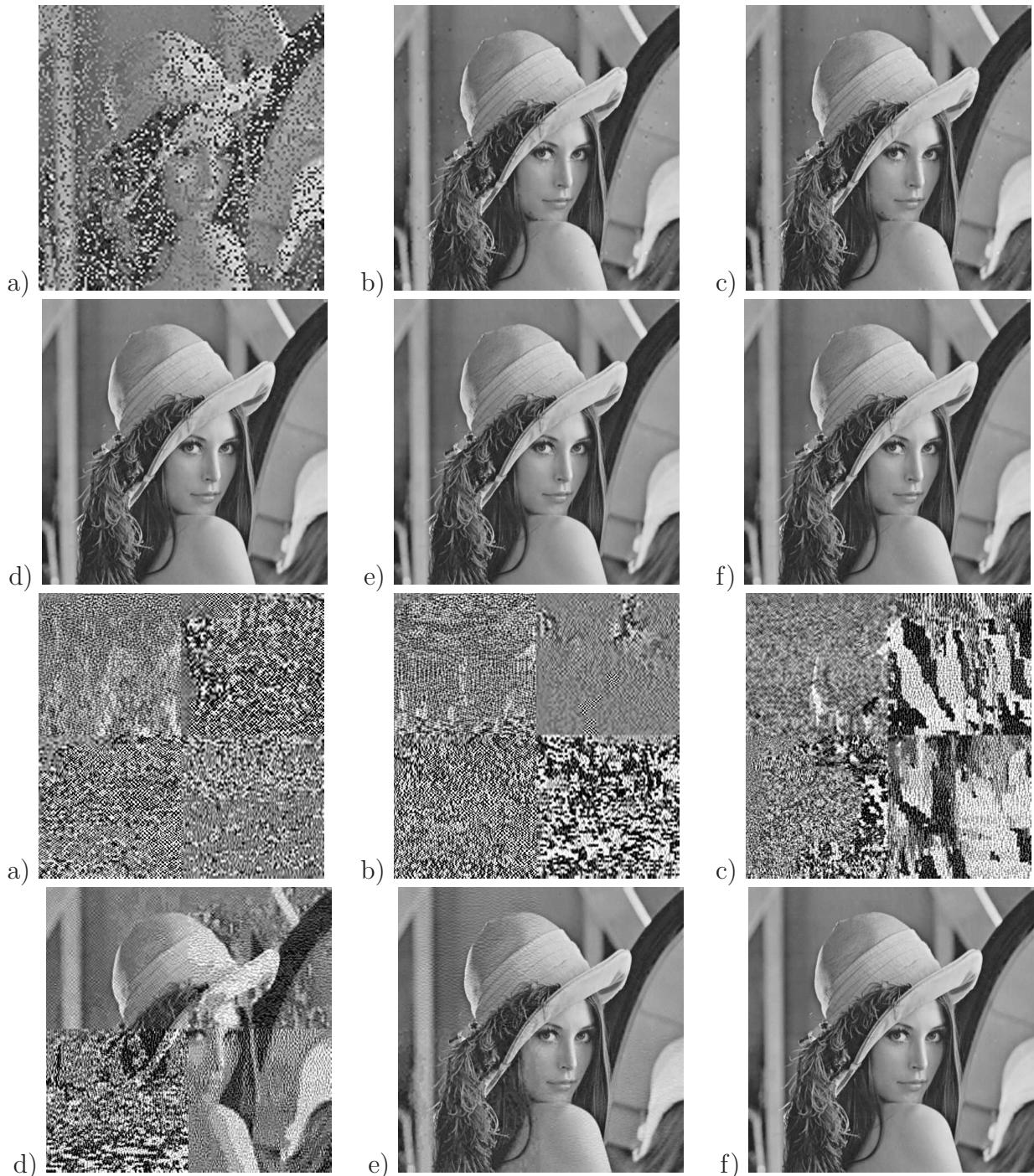


Figure 15: Visual comparison for JSSC (first set) and the SSCC (second set) and the Lena image, for decreasing values of  $\eta_a, \dots, \eta_f$  indicated in Fig. 13 (from left to right).

univerally approached by low-dimensional quantization followed by entropy coding; b) the optimal performance of (almost) lossless joint-source channel coding for a general discrete source and a wide class of channels (BIOS channels in our case) can be attained by linear coding. Our method is *general* in the sense that allows to take advantage of the vast experience developed in lossy compression of practically relevant sources. In fact, the scheme keeps the front-end of state-of-the art source transform coders, comprising a linear projection on a convenient orthonormal basis (such as DWT, short-time DFT, DCT) and scalar quantization, unchanged. Also, the probability models developed in state-of-the art source coders can be reused, provided that these models can be represented in some easily factored form, for the purpose of the joint source-channel Belief Propagation decoder. We developed an example based on JPEG2000 and Turbo Codes. In our case, the probability model reduces to a binary Markov chain for each bit-plane, conditionally on the bits from upper bit-planes. The Markov chains are generally non-stationary. Nevertheless, we showed that there exist binary linear codes that operate arbitrarily close to the Shannon limit for any arbitrary source: a single codebook can handle all sources with given super-entropy. This prompts for a scheme that selects the coding rate for each bit-plane by measuring the *empirical entropy* of the bit-plane itself, based on the probability model used by the decoder, and transmitted to the decoder separately.

The proposed scheme was simulated for some classical test images over a BSC, and compared with a standard separated approach based on JPEG2000 and state-of-the art Turbo Codes. We achieve much better performance in terms of PSNR and, most importantly, in terms of subjective image reconstruction quality. Remarkably, our scheme puts very little stress on the channel coding scheme, and does not require to attain very low BER in order to achieve good end-to-end distortion. We interpret this fact by noticing that using linear codes we eliminate the catastrophic behavior of traditional entropy coding based on Huffman or arithmetic coding, by producing a mapping of the redundant quantization bits over the channel symbols that has a well-conditioned “soft-inverse”. Furthermore, the use of BP decoding allows to exploit the decoder soft-output in order to produce (approximated) MMSE estimates of the transform coefficients using soft-bits.

We would like to conclude by enumerating some topics for further research:

- Several families of linear codes can be used instead of punctured TCs. In particular, instead of using a library of codes to cover a quantized range of rates, we can use *fountain codes* [61, 60] in order to produce the required amount of redundancy “on the fly”, depending on the empirical entropy of the source and, in the case of a compound of BIOS channels whose capacity is known to the transmitter, depending on the capacity of the channel. The use of fountain codes for *universal* lossless

compression was investigated in [23], and a similar encoding scheme can be used in this context. Furthermore, the same approach can be used in multicast applications, where the same source must be sent to several users and each user may have a different channel capacity, as in the classical “fountain coding” setting.

- The representation of the probability models implicitly assumed in various state-of-the art source coders in terms of (conditional) Markov chains is a problem of independent interest. We did not investigate in the details other image coders or coders for different kind of sources, but we believe that an approach similar to that taken here can be applied to a variety of source coders. In particular, Markovian models for state-of-the art speech/audio and video coders may lead to very interesting practical applications.
- The restriction to BIOS channels is made here essentially for simplicity. It is clear that the output of the binary joint-source channel encoder can be concatenated with any suitable modulator in order to drive a non-binary input channel, as commonly done in a “coded modulation” approach. In particular, the binary multistage encoding advocated here to encode the bit-planes can be married to *superposition coding*: instead of sending the encoded bit-planes in sequence, these can be modulated and superimposed. This approach may lead to efficient schemes for multicasting a common source to several users over Gaussian channels in different SNR (or fading) conditions [81, 82, 83, 84, 85].
- From our simulations it appears that the type of reconstruction “noise” achieved by the proposed scheme is very different from the typical effect of non-perfect reconstruction at the output of standard source coders. In fact, in our case it appears to be much closer to some additive independent noise. Hence, our scheme appears to be ideally suited to be concatenated with the DUDE algorithm for denoising [80].

## APPENDIX

### A Proof of Theorem 1

Consider a memoryless stationary BIOS channel with input alphabet  $\mathbb{F}_2$ , output alphabet  $\mathcal{Y}$ , transition probability  $P_{Y|X}(y|x)$ , and capacity  $C$ . Consider also a binary arbitrary source  $V$ , defined by the sequence of  $K$ -th order joint probability distributions  $P_V^{(K)}(\mathbf{v})$  over  $\mathbb{F}_2^K$ , for  $K = 1, 2, \dots$ . Notice that these probabilities need not have any structure, and the source need not be stationary or ergodic. The sup-entropy rate  $\overline{H}(V)$  is defined in [86] as the limsup in probability of the sequence of random variables  $-\frac{1}{K} \log_2 P_V^{(K)}(\mathbf{v})$ , for  $K \rightarrow \infty$ . Theorem 1 is a simple corollary of the following results.

Let  $\mathbf{b} \in \mathbb{F}_2^B$  be a binary vector and  $\mathbf{G}_1 \in \mathbb{F}_2^{B \times N}$  a binary matrix. Consider the linear encoding rule  $\mathbf{c} = \mathbf{b}\mathbf{G}_1$  and some decoding rule  $\psi_1 : \mathcal{Y}^N \rightarrow \mathbb{F}_2^B$ . We define the conditional decoding error probability as

$$\mathbb{P}(e_1|\mathbf{b}) = \mathbb{P}(\psi_1(\mathbf{y}) \neq \mathbf{b} | \mathbf{x} = \mathbf{b}\mathbf{G}_1) \quad (25)$$

Then, we have [12, Problem 11, p. 114]

**Lemma 1.** For every  $\epsilon, \delta > 0$  and sufficiently large  $N$  there exist  $\mathbf{G}_1$  and  $\psi_1$  such that  $\mathbb{P}(e_1|\mathbf{b}) \leq \epsilon, \forall \mathbf{b} \in \mathbb{F}_2^B$ , with  $\frac{B}{N} \geq C - \delta$ .  $\square$

We stated Lemma 1 in terms of the conditional error probability that, due to uniform error property of linear codes, is independent of the information message and therefore coincides with the average and maximal error probabilities. In particular, we let  $\psi_1$  be the Maximum Likelihood decoding rule, defined by

$$\psi_1(\mathbf{y}) = \operatorname{argmax}_{\mathbf{b}' \in \mathbb{F}_2^B} \mathbb{P}(\mathbf{y}|\mathbf{b}'\mathbf{G}_1) \quad (26)$$

Notice that this rule *ignores* the a priori probability of the information messages. Furthermore, the resulting average error probability does not depend on this a priori probability.

Let  $\mathbf{v} \sim P_V^{(K)}$  be a vector of length  $K$  generated by the arbitrary binary source  $V$ , and let  $\mathbf{G}_2 \in \mathbb{F}_2^{K \times B}$  be a binary matrix. Consider the linear encoding rule  $\mathbf{b} = \mathbf{v}\mathbf{G}_2$  and some suitable decoding rule  $\psi_2 : \mathbb{F}_2^B \rightarrow \mathbb{F}_2^K$ . We define the average decoding error probability as

$$\mathbb{P}(e_2) = \sum_{\mathbf{v} \in \mathbb{F}_2^K} \mathbb{P}(\psi_2(\mathbf{b}) \neq \mathbf{v} | \mathbf{b} = \mathbf{v}\mathbf{G}_2) P_V^{(K)}(\mathbf{v}) \quad (27)$$

The following result, which is the key for the proof of Theorem 1, is due to Verdú and Shamai [13] and generalizes a well-known result for memoryless sources [12, Problem 7, p. 24] to sources with an arbitrary statistics:

**Lemma 2.** For any  $\epsilon, \delta > 0$  and sufficiently large  $K$  there exist  $\mathbf{G}_2$  and  $\psi_2$  such that  $\mathbb{P}(e_2) \leq \epsilon$  with  $\frac{B}{K} \leq \overline{H}(V) + \delta$ .  $\square$

Putting together Lemma 1 and 2, consider the following linear coding scheme. Fix  $\epsilon_1, \epsilon_2, \delta_1, \delta_2 > 0$  and choose  $K, B$  and  $N$  such that there exist  $\mathbf{G}_1$  satisfying Lemma 1 with  $\epsilon_1, \delta_1$  and  $\mathbf{G}_2$  satisfying Lemma 2 with  $\epsilon_2, \delta_2$ . We map linearly the source sequence  $\mathbf{v}$  of length  $K$  into the channel codeword  $\mathbf{x} = \mathbf{v}\mathbf{G}_2\mathbf{G}_1$  of length  $N$ . At the receiver, we use a concatenated decoder based on  $\psi_1$  and  $\psi_2$ , that is, we let  $\hat{\mathbf{v}} = \psi_2(\psi_1(\mathbf{y}))$ . Notice that  $\psi_1(\cdot)$  is applied to  $\mathbf{y}$  by ignoring the prior probability on  $\mathbf{v}\mathbf{G}_2$  induced by the source distribution, and  $\psi_2(\cdot)$  is applied on the inner decoding output  $\psi_1(\mathbf{y})$  by ignoring the actual channel output  $\mathbf{y}$ . This corresponds to *separated* decoding and it is, evidently, suboptimal. The error event  $e$  of the concatenated decoder is contained in  $e_1 \cup e_2$ , where  $e_1$  and  $e_2$  are the error events of the inner and outer decoders, respectively. By the union bound we have

$$\mathbb{P}(e) \leq \mathbb{P}(e_1) + \mathbb{P}(e_2) \leq \epsilon_1 + \epsilon_2 = \epsilon \quad (28)$$

The resulting efficiency is given by

$$\eta = \frac{K}{N} \geq \frac{\frac{B}{\overline{H}(V) + \delta_2}}{\frac{B}{C - \delta_1}} \geq \frac{C}{\overline{H}(V)} - \delta \quad (29)$$

for some  $\delta > 0$ . Clearly, both  $\epsilon$  and  $\delta$  vanish as  $\epsilon_1, \epsilon_2, \delta_1$  and  $\delta_2$  vanish. This proves Theorem 1.

Finally, it is clear that the suboptimal concatenated linear scheme with separated two-stage decoding given above cannot perform better than an optimal linear scheme where we design directly an encoding matrix  $\mathbf{G} \in \mathbb{F}_2^{K \times N}$  (not necessarily as given by the product of a “compression” matrix  $\mathbf{G}_2$  and a “channel coding” matrix  $\mathbf{G}_1$ ), and where we use the optimal Maximum a Posteriori (MAP) decoder instead of the concatenation of  $\psi_1$  and  $\psi_2$ . For the sake of completeness, the MAP decoder is given by

$$\begin{aligned} \hat{\mathbf{v}} &= \operatorname{argmax}_{\mathbf{u} \in \mathbb{F}_2^K} \mathbb{P}(\mathbf{u}|\mathbf{y}) \\ &= \operatorname{argmax}_{\mathbf{u} \in \mathbb{F}_2^K} \mathbb{P}(\mathbf{y}|\mathbf{u}\mathbf{G})P_V^{(K)}(\mathbf{u}) \end{aligned} \quad (30)$$

## References

- [1] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [2] J. Kurose and K. Ross, *Computer Networking: A Top-down Approach Featuring the Internet*. Addison-Wesley, 2003.

- [3] *Digital broadcasting systems for television, sound, and data services; Framing structure, channel coding and modulation for digital terrestrial television*, 1997.
- [4] M. Eroz, F. Sun, and L. Lee, “DVB-S2 low density parity check codes with near Shannon limit performance,” *Int. J. Satell. Commun.*, vol. 22, pp. 269–279, 2004.
- [5] M. Gastpar, B. Rimoldi, and M. Vetterli, “To code, or not to code: lossy source-channel communication revisited,” *IEEE Trans. on Inform. Theory*, vol. 49, no. 5, pp. 1147–1158, 2003.
- [6] A. Ortega and K. Ramchandran, “Rate-Distortion Methods for Image and Video Compression: An Overview,” *IEEE Signal Processing Magazine*, vol. 11, 1998.
- [7] J. Ziv, “On universal quantization,” *IEEE Trans. on Inform. Theory*, vol. 31, no. 3, pp. 344–347, 1985.
- [8] R. Zamir and M. Feder, “On universal quantization by randomized uniform/lattice quantizers,” *IEEE Trans. on Inform. Theory*, vol. 38, no. 2, pp. 428–436, 1992.
- [9] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practices*. Kluwer Academic Publishers, 2001.
- [10] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 498–519, February 2001.
- [11] T. Richardson and R. Urbanke, “The capacity of low-density parity check codes under message-passing decoding,” *IEEE Trans. on Inform. Theory*, vol. 47, pp. 599–618, February 2001.
- [12] I. Csiszar and J. Korner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, Inc. Orlando, FL, USA, 1982.
- [13] G. Caire, S. Shamai, and S. Verdu, “Linear Codes for Efficient Data Compression/Transmission,” 2007, Foundations and Trends in Communications and Information Theory, in preparation.
- [14] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo-codes,” *IEEE Trans. on Commun.*, vol. 44, pp. 1261–1271, October 1996.
- [15] A. Aaron and B. Girod, “Compression with side information using turbo codes,” *Proc. of the Data Compression Conference, DCC*, pp. 252–261, 2002.

- [16] J. Garcia-Frias and Y. Zhao, “Compression of binary memoryless sources using punctured turbo codes,” *IEEE Commun. Letters*, vol. 6, no. 9, pp. 394–396, 2002.
- [17] J. Bajcsy and P. Mitran, “Coding for the Slepian-Wolf problem with turbo codes,” *IEEE Global Telecommunications Conference*, vol. 2, 2001.
- [18] J. Hagenauer, J. Barros, and A. Schaefer, “Lossless turbo source coding with decremental redundancy,” in *Proc. International ITG Conference on Source and Channel Coding*, Erlangen, Germany, January 2004, pp. 333–339.
- [19] N. Dütsch and J. Hagenauer, “Combined incremental and decremental redundancy in joint source-channel coding,” in *Proc. International Symposium on Inform. Theory and its Applications*, Parma, Italy, October 2005, pp. 775–779.
- [20] G. Caire, S. Shamai, and S. Verdu, “A new data compression algorithm for sources with memory based on error correcting codes,” *Proc. of the IEEE Information Theory Workshop*, pp. 291–295, 2003.
- [21] ——, “Universal data compression with LDPC codes,” *Third International Symposium On Turbo Codes and Related Topics*, pp. 55–58, 2003.
- [22] ——, “Noiseless data compression with low density parity check codes,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 263–284, 2004.
- [23] G. Caire, S. Shamai, A. Shokrollahi, and S. Verdu, “Fountain Codes for Lossless Data Compression,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2005.
- [24] T. Ancheta Jr, “Syndrome-source-coding and its universal generalization,” *IEEE Trans. on Inform. Theory*, vol. 22, no. 4, pp. 432–436, 1976.
- [25] ——, “Bounds and techniques for linear source coding,” *IEEE Trans. on Inform. Theory*, vol. IT-24, March 1978, Ph.D. thesis abstract.
- [26] J. Massey, “Joint source and channel coding,” in *Proc. of Communication Systems and Random Process Theory, NATO Advanced Studies Institutes Series E25*, Alphen aan den Rijn, The Netherlands, 1978, pp. 279–293, J. K. Skwirzynski editor.
- [27] R. G. Gallager, “Tree encoding for symmetric sources with a distortion measure,” *IEEE Trans. Info. Theory*, vol. 20, pp. 65–76, January 1974.

- [28] R. J. Dick, T. Berger, and F. Jelinek, “Tree encoding of Gaussian sources,” *IEEE Trans. Info. Theory*, vol. 20, pp. 332–335, May 1974.
- [29] M. Hellman, “Convolutional source encoding,” *IEEE Trans. on Inform. Theory*, vol. 21, no. 6, pp. 651–656, 1975.
- [30] A. Viterbi and J. Omura, “Trellis Encoding of memoryless discrete-time sources with a fidelity criterion,” *IEEE Trans. on Inform. Theory*, vol. 20, no. 3, pp. 325–332, 1974.
- [31] V. Koshelev, “Direct sequential encoding and decoding for discrete sources,” *IEEE Trans. on Inform. Theory*, vol. 19, no. 3, pp. 340–343, 1973.
- [32] M. Wainwright and E. Maneva, “Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes,” *Proc. of IEEE Int. Symp. on Inform. Theory ISIT*, pp. 1493–1497, 2005.
- [33] S. Ciliberti, M. Mezard, and R. Zecchina, “Lossy data compression with random gates,” *Arxiv preprint cond-mat/0504509*, 2005.
- [34] E. Martinian and J. Yedidia, “Iterative Quantization Using Codes On Graphs,” *Arxiv preprint cs.IT/0408008*, 2004.
- [35] G. C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, “Turbo codes for non uniform memoryless sources over noisy channel,” *IEEE Trans. on Commun.*, vol. 52, no. 8, pp. 1344–1354, 2004.
- [36] J. Haghighat, W. Hamouda, and M. Soleimani, “Design of lossless turbo source encoders,” *IEEE Signal Processing Letters*, vol. 13, no. 8, pp. 453– 456, Aug. 2006.
- [37] M. Adrat, P. Vary, and J. Spittka, “Iterative source-channel decoder using extrinsic information from softbit-source decoding,” *IEEE ICASSP*, vol. 4, pp. 2653–2656, 2001.
- [38] J. Hagenauer, “Source-controlled channel decoding,” *IEEE Trans. On Commun.*, vol. 43, pp. 2449–2457, September 1995.
- [39] N. Gortz, “On the iterative approximation of optimal joint source-channel decoding,” *IEEE J. Sel. Areas in Commun.*, vol. 19, no. 9, pp. 1662–1670, 2001.
- [40] M. Grangetto, B. Scanavino, and G. Olmo, “Joint Source-Channel Iterative Decoding of Arithmetic Codes,” *IEEE Int. Conf. on Commun., ICC*, vol. 2, pp. 886–890, 2004.

- [41] M. Jeanne, J. Carlach, and P. Siohan, “Joint Source-Channel Decoding of Variable-Length Codes for Convolutional Codes and Turbo Codes,” *IEEE Trans. on Commun.*, vol. 53, no. 1, p. 11, 2005.
- [42] L. Perros-Meilhac and C. Lamy, “Huffman tree based metric derivation for a low-complexity sequential soft VLC decoding,” *IEEE Int. Cong. on Commun., ICC*, vol. 2, 2002.
- [43] ——, “Huffman Tree Based Metric Derivation for a Low-complexity Sequential Soft,” *IEEE Comm. Letters*, vol. 9, no. 10, 2005.
- [44] A. Nosratinia, J. Lu, and B. Aazhang, “Source-channel rate allocation for progressive transmission of images,” *IEEE Trans. on Commun.*, vol. 51, pp. 186–196, February 2003.
- [45] V. Chande and N. Farvardin, “Joint source-channel coding for progressive transmission of embedded source coders,” in *Proceedings IEEE Data Compression Conference*, Snowbird UT USA, March 1999, p. 5261.
- [46] ——, “Progressive transmission of images over memoryless noisy channels,” *IEEE Journal in Selected Areas on Commun.*, vol. 18, pp. 850–860, June 2000.
- [47] V. Stanković, R. Hamzaoui, and Z. Xiong, “Product code error protection of packetized multimedia bitstreams,” in *Proc. ICIP-IEEE International Conference on Image Processing*, 2003.
- [48] V. Stanković, R. Hamzaoui, and D. Saupe, “Fast algorithm for rate-based optimal error protection of embedded codes,” *IEEE Trans. on Commun.*, vol. 51, pp. 1788–1795, November 2003.
- [49] R. Hamzaoui, V. Stanković, and Z. Xiong, “Fast algorithm for distortion-based error protection of embedded image codes,” *IEEE Trans. on Image Processing*, vol. 14, pp. 1417–1421, October 2005.
- [50] A. Torquati, M. Fresia, and F. Lavagetto, “Determination of Optimal Distortion-Based Protection in Progressive Image Transmission: a Heuristic Approach,” in *IEEE Int. Conf. on Commun., ICC*, Glasgow, UK, June 2007.
- [51] R. E. Krichevsky and V. K. Trofimov, “The performance of universal coding,” *IEEE Trans. on Inform. Theory*, vol. IT-27, pp. 199–207, March 1981.

- [52] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Communications of the ACM*, vol. 30(6), pp. 520–540, June 1987.
- [53] J. Rissanen, “Universal coding, information, prediction, and estimation,” *IEEE Trans. Inform. Theory*, vol. 30, no. 4, pp. 629–636, July 1984.
- [54] H. Gish and J. Pierce, “Asymptotically efficient quantizing,” *IEEE Trans. on Inform. Theory*, vol. 14, no. 5, pp. 676–683, 1968.
- [55] A. Gyorgy and T. Linder, “Optimal entropy-constrained scalar quantization of a uniform source,” *IEEE Trans. on Inform. Theory*, vol. 46, no. 7, pp. 2704–2711, 2000.
- [56] D. Marco and D. Neuhoff, “Low-Resolution Scalar Quantization for Gaussian Sources and Squared Error,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 4, pp. 1689–1697, 2006.
- [57] I. Csizár and J. Körner, *Information theory, coding theorems for discrete memoryless system*. Academic, 1981.
- [58] S. Verdu, “A general formula for channel capacity,” *IEEE Trans. on Inform. Theory*, vol. 40, no. 4, pp. 1147–1157, 1994.
- [59] T. Han, *Information-Spectrum Methods in Information Theory*. Springer, 2003.
- [60] O. Etesami and A. Shokrollahi, “Raptor codes on binary memoryless symmetric channels,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 5, pp. 2033–2051, 2006.
- [61] M. Luby, “LT codes,” *Proc. of The 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, 2002.
- [62] R. J. McEliece, *The Handbook of Coding Theory*. Amsterdam, The Netherlands: Elsevier Science Publishers, 1998, ch. Chap 12: The algebraic theory of convolutional codes, pp. 1065–1138.
- [63] R. G. Gallager, “Low density parity check codes,” *IRE Trans. on Inform. Theory*, vol. IT-8, pp. 21–28, January 1962.
- [64] ———, *Low Density Parity Check Codes*, ser. Research monograph series. Cambridge, Mass.: MIT Press, 1963, no. 21.

- [65] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” *Electronics Letters*, vol. 33, no. 6, pp. 457–458, March 1997.
- [66] H. Jin, A. Khandekar, and R. McEliece, “Apri sova based source controlled channel decoding with the itu-t g.729 speech coding standard,” in *in Proc. 2nd International Symposium on Turbo Codes and Related Topics*, Brest, France, September 2000, pp. 1–8.
- [67] A. Roumy, S. Guemghar, G. Caire, and S. Verdu, “Design methods for irregular repeat-accumulate codes,” *IEEE Trans. on Inform. Theory*, vol. IT-27, pp. 1711–1727, August 2004.
- [68] T. Fingscheidt and O. Scheufen, “Robust GSM Speech Decoding Using the Channel Decoder’s Soft Output,” in *Proc. of EUROSPEECH*, Rhodos, Grece, September 1997, pp. 22–25.
- [69] T. Fingscheidt and P. Vary, “Softbit speech decoding: a new approach to error concealment,” *IEEE Trans. on Speech and Audio Proc.*, vol. 9, pp. 240–251, March 2001.
- [70] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Trans. Image Proc.*, vol. 1, pp. 205–220, April 1992.
- [71] D. LeGall and A. Tabatabai, “Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques,” in *Conf. Acoustic, Speech, Signal Processing*, New York, 1988.
- [72] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” *Appl. Comput. Harmon. Anal.*, vol. 3, pp. 186–200, 1996.
- [73] M. Fresia, “Joint Source Channel Coding for Image Transmission over Noisy Channels,” 2006, Ph.D. Thesis, University of Genova.
- [74] H. Ma and J. Wolf, “On tail biting convolutional codes,” *IEEE Trans. on Commun.*, vol. COM-34, no. 2, pp. 104–110, February 1986.
- [75] J. Anderson and S. Hladik, “Tailbiting map decoders,” *IEEE Selected Areas in Commun.*, vol. 16, pp. 297–302, February 1998.
- [76] E. Rosnes and O. Ytrehus, “An efficient algorithm for tailbiting turbo code weight distribution calculation,” in *Proc. of 3rd International Symposium on Turbo-Codes*, Brest, France, September 2003, pp. 439–442.

- [77] S. Shamai and S. Verdu, “The Empirical Distribution of Good Codes,” *IEEE Trans. on Inform. Theory*, vol. 43, no. 3, pp. 836 – 846, 1997.
- [78] L. Bahl, J. Cocke, F. J. J., and Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Trans. on Inform. Theory*, pp. 284–287, March 1974.
- [79] <http://www.openjpeg.org>.
- [80] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, “Universal Discrete Denoising: Known Channel,” *IEEE Trans. on Inform. Theory*, vol. 51, no. 1, p. 5, 2005.
- [81] U. Mittal and N. Phamdo, “Hybrid digital-analog (HDA) joint source-channel codes for broadcasting and robust communications,” *IEEE Trans. on Inform. Theory*, vol. 48, no. 5, pp. 1082–1102, 2002.
- [82] M. Skoglund, N. Phamdo, and F. Alajaji, “Hybrid digital-analog source-channel coding for bandwidth compression/expansion,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 8, pp. 3757–3763, 2006.
- [83] G. Caire and K. Narayanan, “On the SNR exponent of hybrid digital-analog space time coding,” *Allerton Conf., September*, 2005.
- [84] K. Narayanan and G. Caire, “Further Results on the SNR Exponent of Hybrid Digital Analog Space Time Codes,” *Proc. of the UCSD Workshop on Infor. Theory and its Applic., San Diego, CA, Feb*, 2006.
- [85] S. Shamai and A. Steiner, “A Broadcast Approach for a Single-User Slowly Fading MIMO Channel,” *IEEE Trans. on Inform. Theory*, vol. 49, no. 10, p. 2617, 2003.
- [86] T. Han and S. Verdu, “Approximation theory of output statistics,” *IEEE Trans. on Inform. Theory*, vol. 39, no. 3, pp. 752–772, 1993.